

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ &
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

**“ΧΩΡΟΘΕΤΗΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ
ΚΥΚΛΩΜΑΤΩΝ ΜΕ ΜΕΘΟΔΟΥΣ ΤΕΧΝΗΤΗΣ
ΝΟΗΜΟΣΥΝΗΣ”**

Τεμπέλης Αθανάσιος

Βόλος, Οκτώβριος 2019

UNIVERSITY OF THESSALY

**DEPARTMENT OF ELECTRICAL &
COMPUTER ENGINEERING**



**UNIVERSITY OF
THESSALY**

MASTER THESIS

**“DEEP LEARNING FOR LEGALIZING
STANDARD – CELL PLACEMENT”**

By

Tempelis Athanasios

Volos, October 2019

Στην οικογένειά μου

Μεταπτυχιακή Διατριβή για την απόκτηση του Μεταπτυχιακού
διπλώματος Ειδίκευσης «Επιστήμη και Τεχνολογία του Ηλεκτρολόγου
Μηχανικού» του Πανεπιστημίου Θεσσαλίας, στα πλαίσια του
Προγράμματος Μεταπτυχιακών Σπουδών του Τμήματος Ηλεκτρολόγων
Μηχανικών & Μηχανικών Η/Υ του Πανεπιστημίου Θεσσαλίας.

Τεμπέλης Αθανάσιος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός & Μηχανικός Η/Υ
Πανεπιστημίου Θεσσαλίας

Ευχαριστίες

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας, αισθάνομαι την υποχρέωση να ευχαριστήσω ορισμένους από τους ανθρώπους που γνώρισα, συνεργάστηκα μαζί τους και έπαιξαν πολύ σημαντικό ρόλο στην πραγματοποίησή της.

Πρώτο από όλους θέλω να ευχαριστήσω τον επιβλέποντα καθηγητή της διπλωματικής εργασίας, Καθηγητή Σταμούλη Γεώργιο για την πολύτιμη καθοδήγηση του, την εμπιστοσύνη και εκτίμηση που μου έδειξε. Στη συνέχεια θα ήθελα να ευχαριστήσω τον αναπληρωτή καθηγητή Χρήστο Σωτηρίου και τον αναπληρωτή καθηγητή Φώτιο Πλέσσα που δέχτηκαν να είναι μέλη της τριμελούς επιτροπής αξιολόγησης της διπλωματικής εργασίας.

Τέλος, θέλω να ευχαριστήσω την οικογένειά μου, καθώς και τους φίλους μου, που με υπομονή και κουράγιο πρόσφεραν την απαραίτητη ηθική συμπαράσταση όχι μόνο στην ολοκλήρωση της διπλωματικής μου εργασίας αλλά και καθ' όλη τη διάρκεια των σπουδών μου.

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας διατριβής είναι η εκτενής παρουσίαση και ανάλυση του προβλήματος της χωροθέτησης των ολοκληρωμένων κυκλωμάτων και πως μπορεί αυτό να βελτιωθεί με τη χρήση αλγορίθμων τεχνητής νοημοσύνης.

ABSTRACT

The purpose of this thesis is to present and analyze in a comprehensive way the problem of placement integrated circuits and how this can be improved by using artificial intelligence algorithms.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 : ΣΧΕΔΙΑΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ

| | |
|--|-----------|
| 1.1 ΣΧΕΔΙΑΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ | 15 |
| 1.2 ΡΟΗ ΣΧΕΔΙΑΣΗΣ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ | 16 |
| 1.3 ΦΥΣΙΚΗ ΣΧΕΔΙΑΣΗ (PHYSICAL DESIGN) | 19 |
| 1.4 ΕΡΓΑΛΕΙΑ CAD (COMPUTER – AID DESIGN) | 24 |
| 1.5 ΕΡΓΑΛΕΙΑ EDA (ELECTRONIC DESIGN AUTOMATION) | 24 |
| 1.5.1 ΤΟΜΕΙΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ EDA | 24 |

ΚΕΦΑΛΑΙΟ 2 : PLACEMENT

| | |
|--|-----------|
| 2.1 ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ | 27 |
| 2.2 GLOBAL PLACEMENT | 30 |
| 2.2.1 GRAPH PARTITIONING | 30 |
| 2.2.2 ANALYTIC – BASED PARTITIONING | 33 |
| 2.2.2.1 GORDIAN | 33 |
| 2.2.2.2 GORDIAN – L | 40 |
| 2.2.3 FORCED BASED METHODS | 40 |
| 2.2.4 SIMULATED ANNEALING | 41 |
| 2.2.5 CLUSTERING | 41 |
| 2.3 FINAL PLACEMENT | 42 |
| 2.3.1 SIMULATED ANNEALING | 42 |
| 2.3.2 GREEDY APPROACHES | 44 |
| 2.4 LEGALIZATION | 46 |

ΚΕΦΑΛΑΙΟ 3 : ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

| | |
|---|-----------|
| 3.1 ΕΙΣΑΓΩΓΗ | 47 |
| 3.2 ΠΕΡΙΟΧΕΣ ΤΗΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ | 48 |
| 3.3 ΠΡΟΣΕΓΓΙΣΕΙΣ ΓΙΑ ΤΗΝ ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ | 48 |
| 3.4 Η ΕΞΕΛΙΞΗ ΤΗΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ | 49 |

ΚΕΦΑΛΑΙΟ 4 : ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

| | |
|--|-----------|
| 4.1 ΕΙΣΑΓΩΓΗ | 51 |
| 4.2 ΕΙΔΗ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ | 52 |
| 4.3 ΜΑΘΗΣΗ ΜΕ ΕΠΙΒΛΕΨΗ | 52 |
| 4.3.1 ΚΑΝΟΝΕΣ ΤΑΞΙΝΟΜΗΣΗΣ | 53 |
| 4.3.2 ΠΑΡΕΜΒΟΛΗ – ΠΑΛΙΝΔΡΟΜΗΣΗ (REGRESSION) | 53 |
| 4.4 ΜΑΘΗΣΗ ΧΩΡΙΣ ΕΠΙΒΛΕΨΗ | 54 |
| 4.4.1 ΚΟΝΟΝΕΣ ΣΥΣΧΕΤΙΣΗΣ | 54 |
| 4.4.2 ΟΜΑΔΕΣ (CLUSTERS) | 55 |
| 4.5 ΑΛΛΑ ΕΙΔΗ ΜΑΘΗΣΗΣ | 56 |

ΚΕΦΑΛΑΙΟ 5 : ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

| | |
|--|-----------|
| 5.1 ΕΙΣΑΓΩΓΗ | 59 |
| 5.2 ΣΥΝΑΡΤΗΣΕΙΣ ΕΝΕΡΓΟΠΟΙΗΣΗΣ | 60 |
| 5.3 ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ | 61 |
| 5.3.1 ΠΡΟΣΘΙΑ ΤΡΟΦΟΔΟΤΗΣΗ | 62 |
| 5.4 ΑΛΓΟΡΙΘΜΟΙ ΜΑΘΗΣΗΣ ΥΠΟ ΕΠΙΒΛΕΨΗ | 63 |
| 5.4.1 ΚΑΝΟΝΑΣ ΔΕΛΤΑ | 63 |
| 5.4.2 ΑΝΑΣΤΡΟΦΗ ΜΕΤΑΔΟΣΗ ΛΑΘΟΥΣ | 65 |

ΚΕΦΑΛΑΙΟ 1

ΣΧΕΔΙΑΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ

1.1 ΣΧΕΔΙΑΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ

Η σχεδίαση ολοκληρωμένων κυκλωμάτων (integrated circuit design) αποτελεί ένα υποσύνολο του γνωστικού αντικείμενου των Ηλεκτρολόγων Μηχανικών. Η ψηφιακή σχεδίαση ολοκληρωμένων κυκλωμάτων παράγει κυκλωματικά στοιχεία όπως μικροεπεξεργαστές (microprocessors), μνήμες (RAM memories, ROM memories, flash memories), FPGAs και ASICs (Application Specific Integrated Circuits). Τα σύγχρονα ολοκληρωμένα κυκλώματα είναι αρκούντως πολύπλοκα, με ένα chip μεγάλου μεγέθους να αποτελείται πλέον από πάνω από ένα δισεκατομμύριο τρανζίστορ. Το γεγονός αυτό, σε συνάρτηση με την πίεση της αγοράς για παραγωγή ολοκληρωμένων κυκλωμάτων στο μικρότερο δυνατό χρονικό διάστημα έχει οδηγήσει στην ανάπτυξη και χρήση εργαλείων αυτοματοποιημένης σχεδίασης (automated design tools). Η χρήση εργαλείων αυτής της κατηγορίας κρίνεται πλέον απαραίτητη στην πλειοψηφία των βημάτων που ακολουθούνται κατά την σχεδίαση ενός κυκλώματος.

Στην παρακάτω εικόνα παρουσιάζεται η ραγδαία αύξηση του αριθμού των τρανζίστορ σε ορισμένες από τις σημαντικότερες κεντρικές μονάδες επεξεργαστών που έχουν κατασκευαστεί την τελευταία τριακονταετία, σε συνάρτηση με τον νόμο του Moore που υποστηρίζει πως υπάρχει διπλασιασμός του αριθμού των τρανζίστορ που συνθέτουν ένα επεξεργαστή κάθε δύο χρόνια.

Transistor count

curve shows transistor count doubling every two years

Date of introduction

2,600,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
2,300

1971 1980 1990 2000 2011

8004 8080 8085 8090 Z80 MOS 6502 RCA 1802 80386 80486 Pentium AMD K5 AMD K6 Pentium II AMD K6-III AMD K7 Pentium III AMD K8 Barton Atom Core 2 Duo Core i7 (Quad) Six-Core Opteron 2400 8-Core Xeon Nehalem-EX Quad-core z196 8-core POWER7 10-Core Xeon Westmere-EX Six-Core Xeon 7400 Six-Core Core i7 16-Core SPARC T3

Η διαδικασία που ακολουθείται κατά την σχεδίαση ενός IC αποτελείται από
πολλαπλά βήματα, με βασικότερα των οποίων τα ακόλουθα:

- 16

4. *RTL Design*: Περιγραφή της λειτουργίας του κυκλώματος σε επίπεδο καταχωρητών (register-transfer level).
5. *RTL Simulation*: Προσομοίωση της λειτουργίας του κυκλώματος βάσει της περιγραφής που έχει δοθεί στο προηγούμενο βήμα.
6. *Logic Simulation*: Προσομοίωση της λειτουργίας της σχεδίασης με χρήση κατάλληλων λογισμικών.
7. *Floor Planning*: Σχηματική αναπαράσταση μιας «πρώιμης» τοπολογίας όλων των λειτουργικών τμημάτων της σχεδίασης.
8. *Layout*: Αναπαράσταση του ολοκληρωμένου κυκλώματος με μια σειρά γεωμετρικών σχημάτων τα οποία αντιστοιχούν στο σύνολο των στρωμάτων ημιαγωγού, μετάλλου και οξειδίου που απαρτίζουν την σχεδίαση.
9. *Static Timing Analysis*: Μελέτη του χρονισμού του ολοκληρωμένου κυκλώματος, η οποία δεν απαιτεί την εκ νέου προσομοίωση της λειτουργικότητάς του.
10. *Layout Review*: Επανεξέταση του layout που έχουμε σχεδιάσει. Η προκείμενη διαδικασία μπορεί να οδηγήσει στον επανασχεδιασμό του ολοκληρωμένου, μπορεί δηλαδή να λειτουργήσει αναδρομικά και να ανατροφοδοτήσει με επιπλέον πληροφορίες προηγούμενα βήματα της διαδικασίας που έχουν παρέλθει χρονικά.
11. *Design For Test*: Χρήση συγκεκριμένων τεχνικών σχεδίασης οι οποίες προσδίδουν ιδιαίτερα χαρακτηριστικά στο κύκλωμα έτσι ώστε να γίνεται πιο εύκολη η διαδικασία ελέγχου της ορθότητάς του.
12. *Automatic Test Pattern Generation*: Εύρεση της κατάλληλης αλληλουχίας εισόδων, η οποία όταν εφαρμοσθεί βοηθάει στον εντοπισμό τυχών λαθών που μπορεί να υπάρχουν στη συμπεριφορά του κυκλώματος.

13. *Design For Manufacturability*: Εφαρμογή μιας σειράς τεχνικών οι οποίες τροποποιούν κατάλληλα το κύκλωμα έτσι ώστε η υλοποίησή του σε βιομηχανικό περιβάλλον να καθίσταται ευκολότερη.

14. *Mask Data Preparation*: Το βήμα κατά το οποίο η layout περιγραφή του κυκλώματος «μεταφράζεται» σε κάποια καταλληλότερη μορφή η οποία μπορεί να χρησιμοποιηθεί από έναν photomask writer.

15. *Wafer Fabrication*: Η διαδικασία κατά την οποία δημιουργείται το ολοκληρωμένο κύκλωμα, αποκτώντας την μορφή που γνωρίζουμε μελετώντας μια οποιαδήποτε ηλεκτρονική συσκευή.

16. *Packaging*: Το τελευταίο, πρακτικά, στάδιο της κατασκευής. Το κύκλωμα «συσκευάζεται» σε κάποιο κεραμικό ή πλαστικό υλικό προκειμένου να αποφευχθεί η φθορά του, αλλά και να διευκολυνθεί η ηλεκτρική σύνδεσή του και η ενσωμάτωσή του σε ένα ολοκληρωμένο σύστημα.

17. *Device Characterization*: Η διαδικασία κατά την οποία μέσω μιας σειράς μετρήσεων, με την βοήθεια των κατάλληλων οργάνων, συγκεντρώνουμε και παρουσιάζουμε τα ιδιαίτερα χαρακτηριστικά της τελικής υλοποίησης, βάσει ευρέως αποδεκτών μετρικών.

18. *Yield Analysis*: Συλλογή και ανάλυση των κατάλληλων δεδομένων που απαιτούνται για τον εντοπισμό και την διόρθωση αστοχιών που προκύπτουν κατά την λειτουργία του ολοκληρωμένου.

Η παραπάνω αλληλουχία βημάτων δύναται να χωριστεί σε τρία επιμέρους στάδια:

I. System Level Design

II. RTL Design

1.3 ΦΥΣΙΚΗ ΣΧΕΔΙΑΣΗ (PHYSICAL DESIGN)

Η φυσική σχεδίαση ενός ολοκληρωμένου κυκλώματος αποτελείται από τα ακόλουθα βήματα:

1. *Design Netlist*: Ένα netlist σε επίπεδο πυλών (gate level netlist) είναι κατ'ουσίαν το αποτέλεσμα που προκύπτει από την διαδικασία της σύνθεσης (synthesis) ενός ψηφιακού κυκλώματος. Σύνθεση ονομάζεται η διαδικασία κατά την οποία μετατρέπουμε μια σχεδίαση η οποία μας δίνεται σε κάποια γλώσσα περιγραφής υλικού (hardware description language - HDL) σε ένα βέλτιστο netlist σε επίπεδο πυλών το οποίο προσδιορίζεται πλήρως από μια τεχνολογική βιβλιοθήκη. Δύο από τα πιο διαδεδομένα εργαλεία σύνθεσης ψηφιακών κυκλωμάτων είναι ο Synopsys Design Compiler και ο Cadence RTL Compiler.

2. *Floor Planning*: Το floorplanning είναι κατά πολλούς το πρώτο ουσιαστικό βήμα της φυσικής σχεδίασης. Στο βήμα αυτό κάνουμε μια πρώτη εκτίμηση του συνολικού χώρου που απαιτείται για τις βασικές δομικές μονάδες του chip και καθορίζουμε τις σχετικές τους θέσεις εντός του προκειμένου χώρου. Η διαδικασία είναι απαραίτητη διότι μέσω αυτής καθορίζεται το κατά πόσο η προς υλοποίηση σχεδίαση δύναται να χωρέσει εντός της διαθέσιμης περιοχής. Ο πλήρης καθορισμός της περιοχής πραγματοποιείται δυναμικά και μπορεί πολλές φορές να οδηγήσει στον επαναπροσδιορισμό της αρχιτεκτονικής της σχεδίασης.

3. *Partitioning*: Στο βήμα αυτό ο σχεδιαστής καλείται να βρει τον κατάλληλο τρόπο διαίρεσης της περιοχής, που καταλαμβάνει το chip, σε μικρότερες και ευκολότερα διαχειρίσιμες περιοχές. Αυτό γίνεται προκειμένου να διαχωρίσουμε τα functional blocks της σχεδίασης και να διευκολύνουμε περεταίρω την διαδικασία του placement και του routing που περιγράφονται παρακάτω.

4. *Placement*: Η χωροθέτηση πραγματοποιείται σε τέσσερα βήματα, που απώτερο στόχο έχουν την βέλτιστη τοποθέτηση των κελιών της σχεδίασης στον προκαθορισμένο χώρο.

a. *Pre-placement Optimization*: Στη φάση αυτή πραγματοποιούμε βελτιώσεις στην σχεδίασή μας σε επίπεδο netlist πριν αρχίσουμε την τοποθέτηση των κελιών πάνω στην επιθυμητή επιφάνεια (π.χ ελλάτωση του μεγέθους των εμπλεκομένων στη διαδικασία κελιών).

b. *In-Placement Optimization*: Εδώ, πραγματοποιείται επαναβελτιστοποίηση της λογικής της σχεδίασης μας. Το στάδιο αυτό εκτελείται κατ' επανάληψη και συμπεριλαμβάνει διαδικασίες βελτίωσης του κυκλώματος όπως cell sizing, cell moving, cell bypassing, gate duplication, buffer assertion και area recovery.

c. *Post-Placement Optimization*: Μετά την τοποθέτηση των πυλών διορθώνουμε όσες παραβιάσεις έχουν εμφανιστεί (π.χ setup, hold, max trans/cap violations).

d. *Post-Placement Optimization after Clock Tree Synthesis*: Στην τελευταία φάση της διαδικασίας του placement βελτιστοποιούμε τον χρονισμό του κυκλώματος και προσπαθούμε να διατηρήσουμε το clock skew.

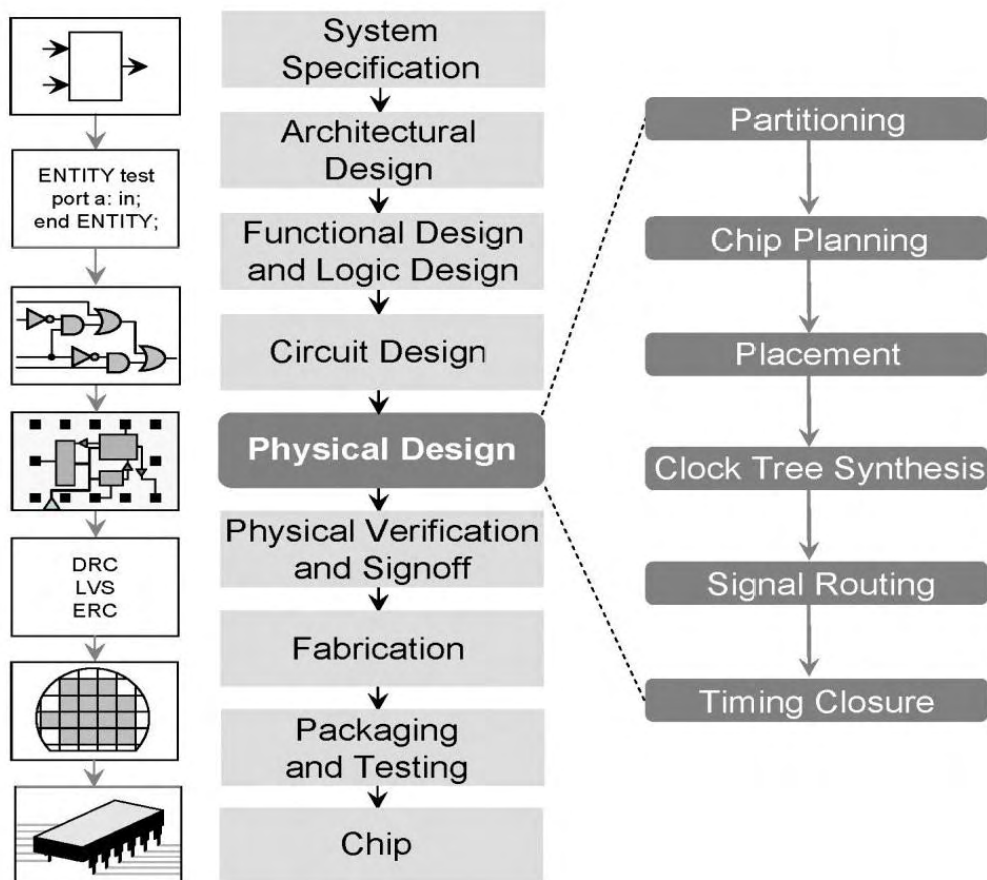
5. *Clock Tree Synthesis*: Ο στόχος του clock tree synthesis είναι να ελαχιστοποιήσουμε το skew και το insertion delay. Εάν το ρολόι της σχεδίασής μας διαμερίζεται σε πολλαπλά blocks τότε χρειάζεται επιμέρους skew analysis.

6. *Routing*: Υπάρχουν δύο τύποι routing, το global routing και το detailed routing. Το global routing τοποθετεί τα routing resources τα οποία χρησιμοποιούνται για τις συνδέσεις μεταξύ των κελιών, ενώ το detailed routing αναθέτει συγκεκριμένα μονοπάτια (routes) σε συγκεκριμένα επίπεδα μετάλλου.

7. *Physical Verification*: Στο τελευταίο στάδιο της φυσικής σχεδίασης ελέγχουμε την ορθότητα του layout που έχει παραχθεί. Αυτό περιλαμβάνει:

- a. Την πλήρη συμμόρφωση της σχεδίασής μας με όλες τις τεχνολογικές απαιτήσεις που έχουν τεθεί (Design Rule Checking - DRC).
- b. Την συνέπεια ως προς το αρχικό netlist (Layout vs. Schematic - LVS).
- c. Την εξάλειψη antenna effects (Antenna Rule Checking).
- d. Την πλήρη συμφωνία της σχεδίασης μας με τις ηλεκτρικές απαιτήσεις που έχουν τεθεί (Electrical Rule Checking - ERC).

Στην παρακάτω εικόνα παρουσιάζεται μια τυπική ροή σχεδίασης ενός VLSI κυκλώματος, με ιδιαίτερη έμφαση στο στάδιο της φυσικής σχεδίασης.



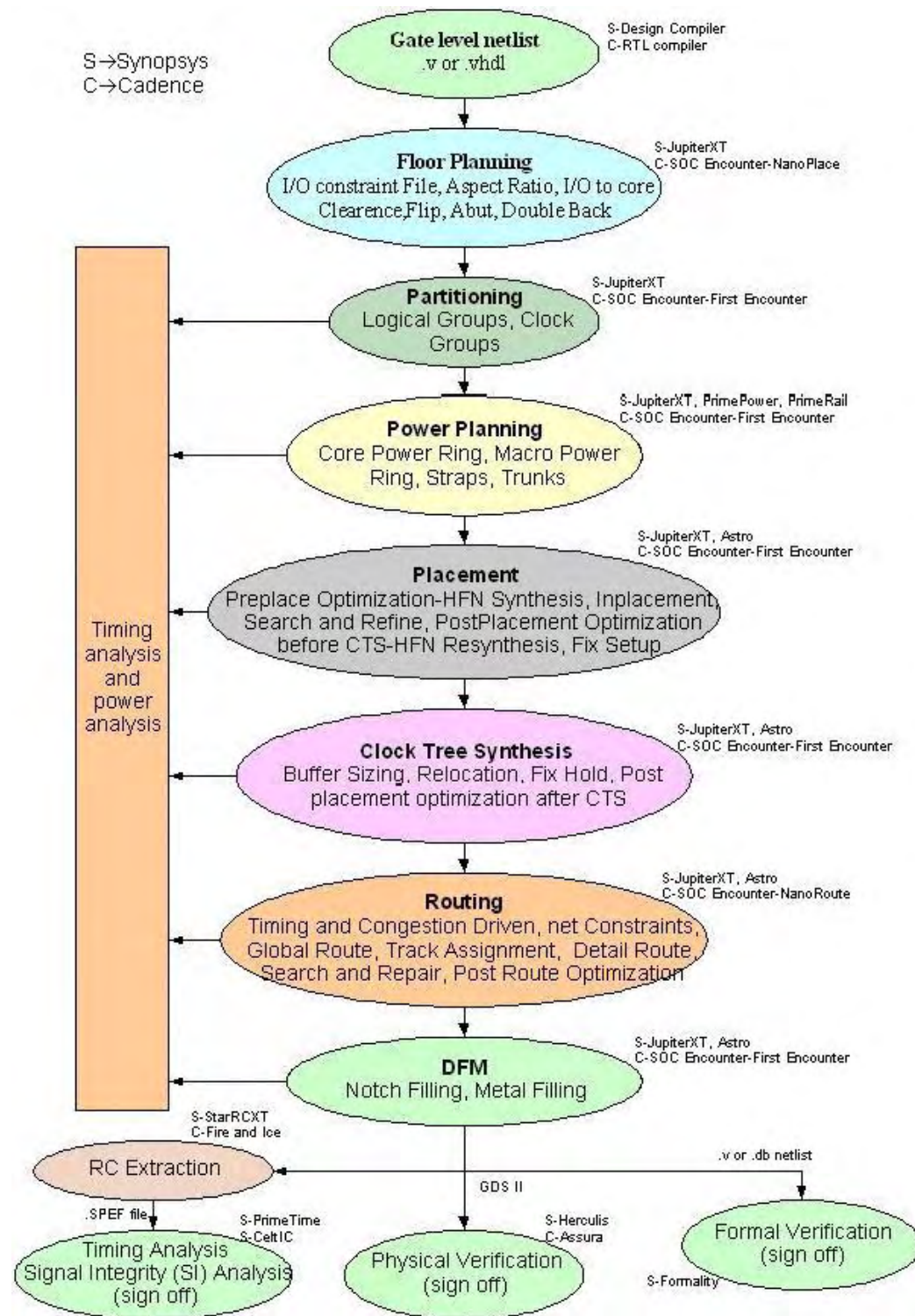
Τα βήματα που περιγράψαμε παραπάνω είναι ουσιαστικά μια αφαιρετική περιγραφή του τελευταίου σταδίου σχεδίασης ενός ολοκληρωμένου κυκλώματος. Πιο αναλυτικές ροές σχεδίασης μπορούν να προκύψουν ανάλογα με τα λογισμικά που χρησιμοποιούνται και την επιλογή των κατάλληλων μεθοδολογιών. Μερικά από τα πιο ευρέως διαδεδομένα λογισμικά που χρησιμοποιούνται στην βιομηχανία είναι τα ακόλουθα:

- Cadence (SoC Encounter, NanoRoute)
- Synopsys (Design Compiler)
- Magma (Blast Fusion)
- Mentor (Olympus SoC, Calibre)

Μια πιο λεπτομερής ροή της φυσικής σχεδίασης παρουσιάζεται στην παρακάτω εικόνα, ενώ παράλληλα αναφέρονται τα ακριβή βήματα και τα εργαλεία που χρησιμοποιούνται σε κάθε ένα από αυτά.

Στο σημείο αυτό πρέπει να αναφέρουμε πως κατά την διαδικασία σχεδίασης ενός ολοκληρωμένου κυκλώματος δύναται να ακολουθηθούν διαφορετικές ροές σχεδίασης (είτε πιο λεπτομερείς είτε πιο γενικευμένες), ανάλογα με την σημαντικότητα και τη βαρύτητα που δίνεται σε κάθε ένα από τα παραπάνω βήματα.

Η παρουσίαση της παρακείμενης ροής σχεδίασης, με χρήση των συγκεκριμένων εργαλείων, γίνεται λόγω της ευρείας χρήσης της από την πλειοψηφία των, σχετιζόμενων με τον κλάδο, του σχεδιασμού ολοκληρωμένων κυκλωμάτων, εταιρειών.



1.4 ΕΡΓΑΛΕΙΑ CAD (COMPUTER-AID DESIGN)

Όταν αναφερόμαστε στο όρο Computer-Aided Design (CAD), αναφερόμαστε πρακτικά στην χρησιμοποίηση της τεχνολογίας των υπολογιστών για την διευκόλυνση της διαδικασίας σχεδιασμού ενός αντικειμένου.

Τα προγράμματα CAD πρωτοεμφανίστηκαν την δεκαετία του '80 και οδήγησαν στο να μειωθεί η άμεση ανάγκη επιπλέον εξειδικευμένου προσωπικού, δίνοντας περεταίρω δυνατότητες ανάπτυξης σε εταιρείες μεσαίου μεγέθους. Το γεγονός πως, σε σύντομο σχετικά χρονικό διάστημα, από την πρώτη εμφάνισή τους, οι εταιρείες κατέστησαν οικονομικά προσιτή την απόκτησή τους και την χρήση τους στον προσωπικό υπολογιστή ενός μέσου χρήστη, βοήθησε στο μέγιστο βαθμό τους σχετιζόμενους με το αντικείμενο κλάδους μηχανικής.

1.5 ΕΡΓΑΛΕΙΑ EDA (ELECTRONIC DESIGN AUTOMATION)

Η αυτοματοποιημένη ηλεκτρονική σχεδίαση (Electronic Design Automation - EDA), είναι κατ'ουσίαν, η υλοποίηση ενός κυκλώματος με παράλληλη χρήση ειδικών λογισμικών τα οποία έχουν δημιουργηθεί κατά περίπτωση για την υποβοήθηση της διαδικασίας. Στην περίπτωση που εξετάζουμε, στον σχεδιασμό, δηλαδή, ενός ολοκληρωμένου κυκλώματος, η εν σειρά χρησιμοποίηση μιας ομάδας εργαλείων για την παραγωγή της τελικής σχεδίασης στοιχειοθετεί μια ροή σχεδιασμού (design flow).

1.5.1 ΤΟΜΕΙΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ EDA

Οι βασικότεροι τομείς της σχεδίασης ενός ολοκληρωμένου κυκλώματος, για τους οποίους έχουν αναπτυχθεί κατάλληλα λογισμικά τα οποία εμπίπτουν στην κατηγορία των EDA εργαλείων είναι οι ακόλουθοι:

➤ ΣΧΕΔΙΑΣΜΟΣ (DESIGN)

- High Level Synthesis
- Logic Synthesis
- Schematic Capture
- Layout

➤ ΠΡΟΣΟΜΟΙΩΣΗ (SIMULATION)

- Logic Simulation
- Behavioral Simulation
- Hardware Emulation

➤ ΑΝΑΛΥΣΗ ΚΑΙ ΕΠΑΛΗΘΕΥΣΗ (ANALYSIS & VERIFICATION)

- Functional Verification
- Formal Verification
- Equivalence Checking
- Static Timing Analysis
- Physical Verification

➤ ΚΑΤΑΣΚΕΥΗ (MANUFACTURING)

- Mask Data Preparation
 - Mask Generation
 - Automatic Test Pattern Generation
 - Built-In Self-Test

ΚΕΦΑΛΑΙΟ 2

PLACEMENT

Στο κεφάλαιο αυτό θα παρουσιάσουμε αναλυτικά το θεωρητικό υπόβαθρο του προβλήματος του Placement και τις μελέτες που έχουν γίνει στο συγκεκριμένο ζήτημα έως σήμερα.

2.1 ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Η διαδικασία του Placement μπορεί να οριστεί ως το πρόβλημα χωροθέτησης (εύρεσης του κατάλληλου τρόπου τοποθέτησης) μιας ομάδας αντικειμένων σε έναν πλήρως καθορισμένο χώρο, προκειμένου να ελαχιστοποιηθεί η τιμή μιας αντικειμενικής συνάρτησης. Στην περίπτωση που καλούμαστε να τοποθετήσουμε «ιδανικά» ένα σύνολο κελιών τα οποία αποτελούν ένα ολοκληρωμένο κύκλωμα, η αντικειμενική αυτή συνάρτηση αναφέρεται, στην πλειοψηφία των περιπτώσεων, είτε στο συνολικό μήκος καλωδίου που χρησιμοποιείται για τη σύνδεση των κελιών, είτε στην συνολική επιφάνεια που αυτά καταλαμβάνουν.

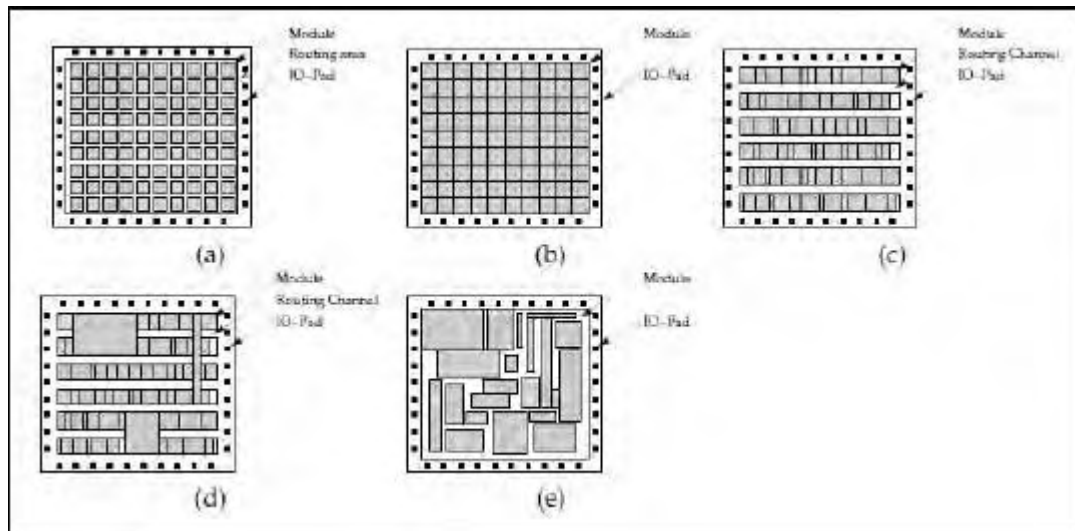
Το τελικό αποτέλεσμα που προκύπτει μετά την ολοκλήρωση της προκειμένης διαδικασίας αποτυπώνεται σε μορφή layout. Υπάρχουν πέντε βασικές κατηγορίες layout οι οποίες διαχωρίζονται, κατά κύριο λόγο, από την ελευθερία που παρέχεται στον σχεδιαστή κατά την υλοποίηση του τελικού αποτελέσματος:

- a. Gate Array
- b. Sea Of Gates
- c. Standard Cell

d. Mixed Cell

e. General Cell (Macros)

Στην εικόνα που ακολουθεί παρουσιάζονται ενδεικτικά παραδείγματα των παραπάνω κατηγοριών.



Στη διεθνή βιβλιογραφία το πρόβλημα του Placement διαιρείται σε πέντε υποκατηγορίες - βήματα:

1. *Global Placement*: Το global placement παράγει ένα αρχικό placement στο οποίο υπάρχει αλληλεπικάλυψη (overlap) μεταξύ των κελιών. Η διαδικασία του global placement μπορεί να εκτελεστεί επαναληπτικά, προκειμένου να παραχθεί κάποιο καλύτερο αποτέλεσμα.

2. *Final Placement*: Το final placement βελτιστοποιεί τις θέσεις των κελιών που έχουν προκύψει από το παραπάνω βήμα. Η διαδικασία είναι πάντοτε επαναληπτική και πρακτικά δίνει αποτελέσματα τα οποία κινούνται εντός ενός συγκεκριμένου συνόλου λύσεων. Η τελική λύση που προκύπτει είναι, συνήθως, μια συγκεκριμένη χωροθέτηση των κελιών χωρίς κανένα φαινόμενο επικάλυψης.

3. *Area Minimization*: Το πρόβλημα ελαχιστοποίησης της καταλαμβανόμενης από τα κελιά περιοχής είναι κατά βάση ένα δισδιάστατο πρόβλημα τοποθέτησης αντικειμένων σε περιορισμένο χώρο και ανήκει στα NP-hard προβλήματα.

4. *Legalization*: Εάν το τελικό κύκλωμα εξακολουθεί να παρουσιάζει αλληλεπικάλυψη μεταξύ των κελιών, πρέπει εκ νέου να εφαρμοστούν τεχνικές «νομιμοποίησης» των θέσεών τους.

5. *Post-Placement Optimization*: Σε ορισμένες περιπτώσεις η τελική λύση του προβλήματος μπορεί να βελτιωθεί αισθητά, εφαρμόζοντας επιπλέον τεχνικές οι οποίες αφήνουν ανέπαφη την πλειοψηφία των κελιών.

2.2 GLOBAL PLACEMENT

Εάν ανατρέξει κάποιος στη σύγχρονη βιβλιογραφία για το ζήτημα του global placement θα παρατηρήσει πως οι προσεγγίσεις που έχουν παρουσιαστεί μέχρι και σήμερα είναι ελάχιστες. Οι βασικότερες από αυτές είναι ο διαχωρισμός του κυκλώματος βάσει του γράφου διασυνδεσιμότητας των στοιχείων του (graph partitioning) και ο διαχωρισμός με χρήση αναλυτικών μεθόδων (analytic based placement). Στα κεφάλαια που ακολουθούν θα παρουσιαστούν οι προαναφερθείσες προσεγγίσεις ενώ, επιπρόσθετα θα γίνει αναφορά και σε κάποιες σπανιότερα εφαρμοζόμενες τεχνικές.

2.2.1 GRAPH PARTITIONING

Η συγκεκριμένη προσέγγιση βασίζεται στην εύρεση του κατάλληλου τρόπου διαχωρισμού του υπεργράφου (hypergraph) που σχηματίζεται εάν λάβουμε υπόψη μας όλες τις συνδέσεις μεταξύ των στοιχείων που απαρτίζουν το προς εξέταση κύκλωμα.

Η διεργασία διαχωρισμού (partitioning) εκτελείται αναδρομικά και τμηματοποιεί την σχεδίαση βάσει προεπιλεγμένων κριτηρίων.

Στην πλειοψηφία των περιπτώσεων βασικό κριτήριο αποτελεί η ισόποση κατανομή των βαρών των ακμών του γράφου στις δύο νέες περιοχές που δημιουργούνται. Όταν αναφερόμαστε στο βάρος μιας ακμής αναφερόμαστε κατ'ουσίαν σε μια τιμή που της έχει ανατεθεί σε σχέση με την βαρύτητα που παρουσιάζει ως προς την σημαντικότητα της σύνδεσης των προκείμενων κελιών. Ο τρόπος υπολογισμού έχει άμεση σχέση με το net model που χρησιμοποιείται για την αναπαράσταση του κυκλώματος σε μορφή γράφου. Τα σημαντικότερα μοντέλα είναι τα ακόλουθα:

1. Distance Based: Για τον προσδιορισμό του μήκους καλωδίου του εκάστοτε net υπολογίζεται το άθροισμα της ευκλείδειας απόστασης μεταξύ όλων εκείνων των στοιχείων του κυκλώματος που συνδέονται μεταξύ τους και το απαρτίζουν.

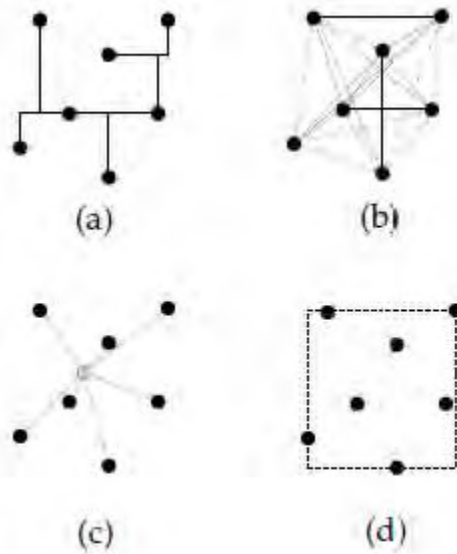
2. Steiner Tree: Στην πλειοψηφία των αλγορίθμων που χρησιμοποιούν αυτή τη μορφή αναπαράστασης των net παρουσιάζεται επιπρόσθετα και μια επέκταση αυτού του μοντέλου το ευθύγραμμο (rectilinear) Steiner Tree. Η χρήση ενός rectilinear Steiner Tree παρουσιάζει σημαντικά πλεονεκτήματα εάν αναλογιστούμε πως την διαδικασία του placement διαδέχεται η διαδικασία του routing. Ένα δέντρο της προκειμένης μορφής αποτελείται αποκλειστικά από οριζόντιες και κάθετες ακμές οι οποίες εκτείνονται προς εκείνες τις κατευθύνσεις που απαιτείται για να συμπεριληφθούν όλα τα εμπλεκόμενα στο net στοιχεία. Ο αριθμός των ακμών είθισται να είναι ο μικρότερος δυνατός βάσει του οποίου ελαχιστοποιείται το συνολικό μήκος καλωδίου που απαιτείται για την διασυνδεσιμότητα των κελιών και των pins που αποτελούν το net.

3. Clique: Η χρήση του μοντέλου κλίκας είναι αρκούντως διαδεδομένη στα εργαλεία που υλοποιούν αλγορίθμους global placement. Το συγκεκριμένο μοντέλο βασίζεται στην κατάλληλη μετατροπή ενός net σε ένα υπογράφο κλίκας. Για τον υπολογισμό του βάρους των ακμών που συνθέτουν το net χρησιμοποιείται ο μαθηματικός τύπος : $w(i,j)=2/k$. Το βάρος, δηλαδή, μιας υπάρχουσας ακμής που προσδιορίζει την σύνδεση μεταξύ ενός στοιχείου i και ενός στοιχείου j ισούται με το αποτέλεσμα της διαίρεσης του δύο με το συνολικό αριθμό στοιχείων που απαρτίζουν την κλίκα. Ο υπολογισμός γίνεται βάσει του παραπάνω τύπου προκειμένου να ελαττώσουμε, χωρίς να διαταράξουμε την συνολική, επιρροή των μεγάλων δικτυωμάτων έναντι των μικρότερων που αποτελούν συνήθως την πλειοψηφία σε μια μέση ψηφιακή σχεδίαση. Βασικό χαρακτηριστικό και πλεονέκτημα του μοντέλου είναι η μικρή πολυπλοκότητα υπολογισμού των βαρών.

4. Star: Το μοντέλο αυτό είναι μια παραλλαγή του μοντέλου κλίκας, όπου οι υπερακμές του σχηματιζόμενου υπεργράφου, μετατρέπονται σε υπογράφους σε σχηματισμό αστέρα.

5. Bounding Box: Ο υπολογισμός του συνολικού μήκους καλωδίου γίνεται βάσει της ημιπεριμέτρου του μικρότερου παραλληλόγραμμου που περιβάλλει κάθε net.

Στην παρακάτω εικόνα παρουσιάζονται χαρακτηριστικά παραδείγματα των παραπάνω μοντέλων.



2.2.2 ANALYTIC – BASED PARTITIONING

Η χρήση αναλυτικών μεθόδων για το κατάλληλο partitioning του κυκλώματος σχετίζεται άμεσα με το graph partitioning. Η βασική τους, όμως, διαφορά έγκειται στο ότι ο βασικός στόχος, πλέον, δεν είναι η εύρεση του min cut, αλλά ο προσδιορισμός της καλύτερης δυνατής θέσης για το cut line. Ο σημαντικότερος αλγόριθμος αυτής της κατηγορίας είναι ο Gordian. Στα κεφάλαια που ακολουθούν θα παρουσιαστεί τόσο ο προαναφερθείς αλγόριθμος όσο και μια από τις βασικότερες παραλλαγές του ο Gordian-L.

2.2.2.1 GORDIAN

Ο αλγόριθμος Gordian έχει δύο βασικά χαρακτηριστικά, πραγματοποιεί global optimization σε κάθε βήμα και κατά το partitioning δημιουργεί αποκλειστικά ορθογώνιες περιοχές.

Συνοπτικά τα βήματα του αλγορίθμου είναι τα εξής:

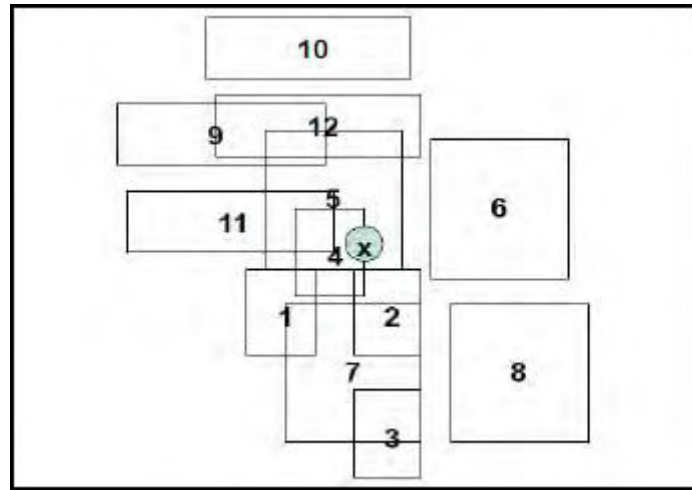
- Τοποθετούμε όλα τα στοιχεία του κυκλώματος εντός της προβλεπόμενης περιοχής που έχουμε στη διάθεσή μας.
- Χρησιμοποιούμε κατάλληλες μεθόδους τμηματοποίησης για να δημιουργήσουμε ομάδες στοιχείων ενώ παράλληλα ακολουθούμε ρητά τους κανόνες καθολικής βελτιστοποίησης που έχουν τεθεί.
- Όταν οι ομάδες στοιχείων που έχουν σχηματιστεί έχουν πλήθος στοιχείων μικρότερο από μια μεταβλητή, η τιμή της οποίας έχει προαποφασιστεί στην αρχή της εκτέλεσης του αλγορίθμου, τοποθετούμε καταλλήλως τα στοιχεία στον διαθέσιμο χώρο.

Η συνολική λειτουργία του αλγορίθμου μπορεί να αποτυπωθεί υπό την μορφή ψευδοκώδικα, μία τυποποιημένη εκδοχή του οποίου παρουσιάζεται στην εικόνα που ακολουθεί.

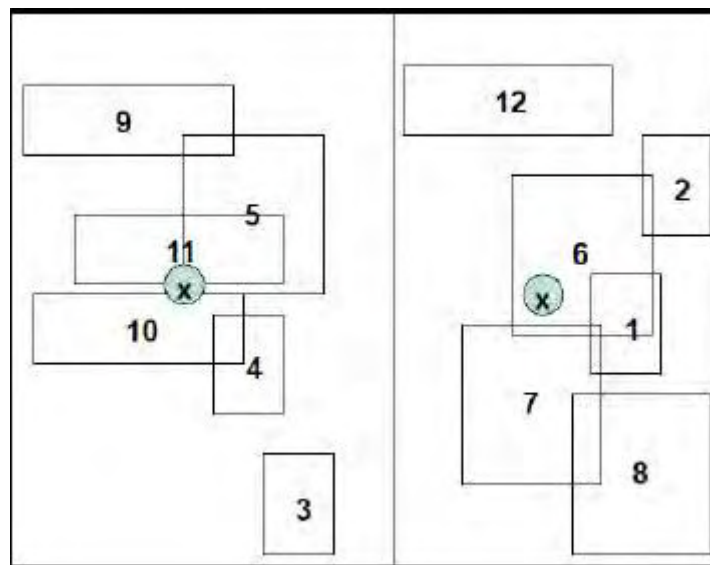
```
Gordian Procedure:  
l := 1  
global_optimize( l );  
while ( there exists  $|M_l| > K$  )  
    for each r  
        partition( r, r', r'' );  
    l ++;  
    setup_constraints( l );  
    global_optimize( l );  
    re-partition( l );  
final_placement( l );  
end_procedure;
```

όπου l είναι το προκείμενο βήμα εκτέλεσης και r είναι το προς εξέταση partition.

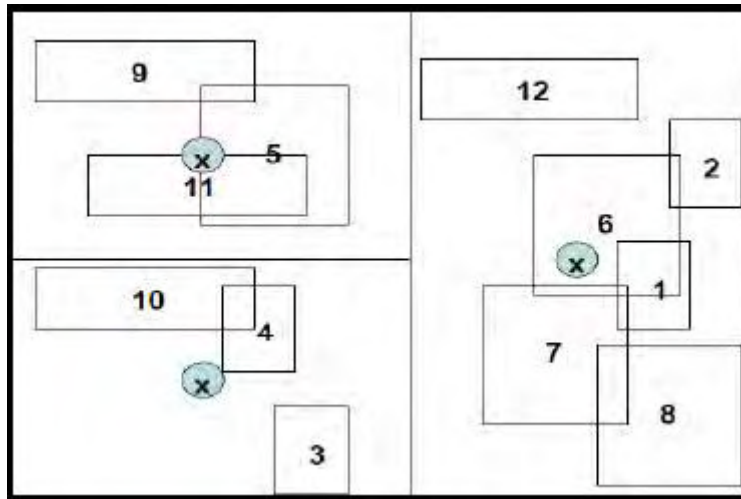
Στις εικόνες που ακολουθούν παρουσιάζεται ένα απλό παράδειγμα εφαρμογής του αλγορίθμου, για την χωροθέτηση ενός κυκλώματος έντεκα πυλών με ελάχιστο αριθμό κελιών ανά εξεταζόμενη περιοχή ίσο με τρία.



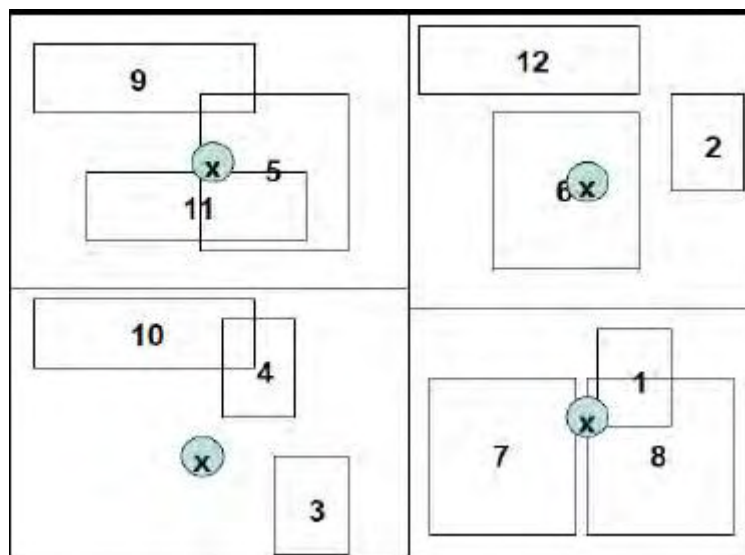
Τυχαία αρχική τοποθέτηση των κελιών



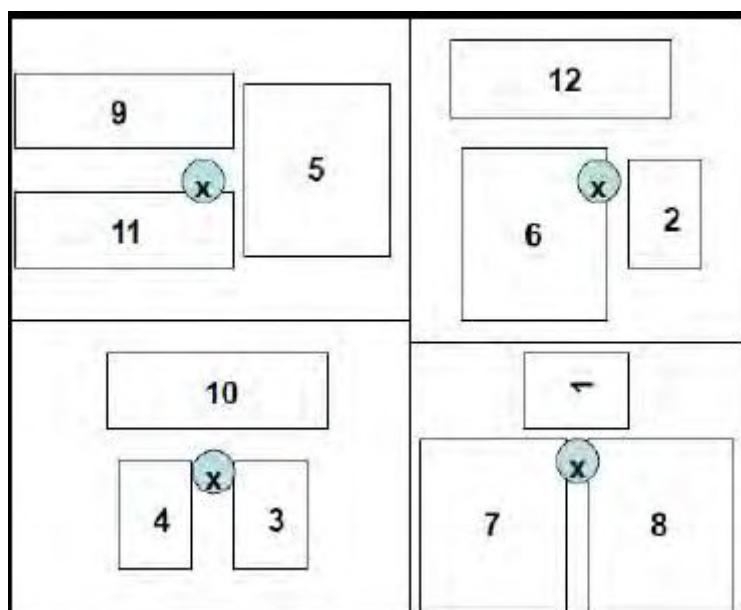
1^ο partitioning και αναπαραγωγή των θέσεων



2° partitioning και αναπαραγωγή των θέσεων



3° partitioning και αναπαραγωγή των θέσεων



4^ο partitioning και αναπαραγωγή των θέσεων

Στο σημείο αυτό θα διατυπώσουμε αναλυτικότερα την ακολουθία των βημάτων που έχουν αναφερθεί παραπάνω. Αρχικά, λοιπόν, χρησιμοποιώντας το clique net model σχηματίζουμε έναν υπεργράφο ο οποίος αναπαριστά, λεπτομερώς, τον τρόπο σύνδεσης των στοιχείων που απαρτίζουν το κύκλωμα. Για τον υπολογισμό του βάρους της εκάστοτε ακμής χρησιμοποιείται ο τύπος $2/K$ που έχει αναφερθεί σε προηγούμενη ενότητα. Επιπρόσθετα, τοποθετούμε τα pin της σχεδίασης στις κατάλληλες θέσεις (μια σύμβαση που ακολουθείται στα περισσότερα κυκλώματα είναι να τοποθετούμε τα input pins στην αριστερή και στην πάνω πλευρά του παραλληλογράμμου που αναπαριστά τον χώρο που έχουμε στη διάθεσή μας και τα output pins στην κάτω και στην δεξιά πλευρά αυτού).

Ακολουθώντας, προχωρούμε στον υπολογισμό μιας σειράς πινάκων οι οποίοι θα μας βοηθήσουν, δυνητικά, στην μαθηματική διατύπωση του προκείμενου προβλήματος. Ο πρώτος πίνακας που υπολογίζουμε είναι ο πίνακας διασυνδεσιμότητας κελιών (adjacency matrix), ο οποίος έχει διαστάσεις $n \times n$, όπου n ο αριθμός των κελιών του κυκλώματος. Κάθε στοιχείο του πίνακα έχει την τιμή του βάρους της ακμής που συνδέει τα κελιά που προσδιορίζονται από την θέση (γραμμή, στήλη) του στοιχείου. Η μη ύπαρξη ακμής στον γράφο συνεπάγεται την τοποθέτηση μηδενικής τιμής στην, προς συμπλήρωση, θέση του πίνακα.

Ο επόμενος πίνακας που υπολογίζουμε είναι ο pin connection matrix, ο πίνακας, δηλαδή, που περιγράφει τις συνδέσεις των κελιών με τα input και output pins. Οι διαστάσεις του είναι $n \times m$ όπου m ο συνολικός αριθμός των pins. Ο τρόπος υπολογισμού της κάθε θέσης του πίνακα γίνεται με όμοιο τρόπο όπως με τον προηγούμενο πίνακα.

Στη συνέχεια, υπολογίζεται ο degree matrix. Όλα τα στοιχεία του $n \times n$ πίνακα είναι μηδενικά εκτός από τα στοιχεία της κύριας διαγωνίου. Κάθε ένα από αυτά προκύπτει προσθέτοντας όλες τις τιμές των αντίστοιχων γραμμών των πινάκων adjacency matrix και pin connection matrix.

Αφού ολοκληρώσουμε, τον παραπάνω υπολογισμό, διαμορφώνουμε τον laplacian matrix, ο οποίος χρησιμοποιείται άμεσα κατά την μορφοποίηση του μαθηματικού προβλήματος που καλούμαστε να λύσουμε. Ο συγκεκριμένος πίνακας είναι το αποτέλεσμα της αφαίρεσης του adjacency matrix από τον degree matrix.

Στο επόμενο βήμα διαμορφώνουμε τα fixed pin vectors. Τα fixed pins vectors εξαρτώνται από το pin connection matrix και την θέση των pins στην περιφέρεια της, προς χρήση, συνολικής περιοχής. Ο μαθηματικός τύπος υπολογισμού κάθε θέσης του διανύσματος είναι ο ακόλουθος: $dx_i = -\sum p_{ij} \cdot x(p_j)$, όπου p_{ij} είναι η τιμή της αντίστοιχης θέσης του pin connection matrix και $x(p_j)$ η συντεταγμένη x του pin j . Με τον ίδιο τρόπο υπολογίζουμε και το αντίστοιχο διάνυσμα για την κατεύθυνση y .

Στο σημείο αυτό μπορούμε να διατυπώσουμε το quadratic problem που καλούμαστε να επιλύσουμε. Θέτουμε ως (x_i, y_j) τις συντεταγμένες του κέντρου ενός κελιού i , w_{ij} το βάρος της ακμής που ενώνει δύο κελιά i και j , και \mathbf{x}, \mathbf{y} , τα διανύσματα που θα αποθηκευτούν οι τελικές συντεταγμένες. Το κόστος της προκείμενης σύνδεσης

δίνεται από τον ακόλουθο τύπο: $0.5 * w_{ij} * ((x_i - x_j)^2 + (y_i - y_j)^2)$, και βάσει αυτού προκύπτει η συνάρτηση που περιγράφει το συνολικό κόστος:

$$F(x, y) = 0.5 * \mathbf{x}^T * L * \mathbf{x} + d_{xT} * \mathbf{x} + 0.5 * \mathbf{y}^T * L * \mathbf{y} + d_{yT} * \mathbf{y} + const$$

όπου \mathbf{x}, \mathbf{y} είναι τα διανύσματα που εσωκλείουν τις συντεταγμένες όλων των κελιών του κυκλώματος, L ο laplacian matrix, τον τρόπο υπολογισμό του οποίου περιγράψαμε παραπάνω, d_x, d_y τα fixed pins vectors και $const$ μια σταθερά η τιμή της οποίας υπολογίζεται δυναμικά μέσα από αλληπάλληλες πειραματικές εκτελέσεις του αλγορίθμου.

Το ζητούμενο πλέον είναι η ελαχιστοποίηση της παραπάνω συνάρτησης. Υπάρχουν, όμως, επιπλέον περιορισμοί τους οποίους πρέπει να λάβουμε υπόψη. Κατά την δημιουργία των νέων partition, πρέπει τα κέντρα των νέων περιοχών να «συγκρατούν» εντός της υφιστάμενης περιοχής τα αντίστοιχα κελιά. Η μαθηματική διατύπωση του περιορισμού είναι η ακόλουθη:

$$A^{(i)} * \mathbf{x} = \mathbf{u}^{(i)}$$

Αντίστοιχα προσδιορίζεται ο περιορισμός και για την έτερη κατεύθυνση. Στο στάδιο i , λοιπόν, της βελτιστοποίησης πρέπει το διάνυσμα των συντεταγμένων των κελιών να επιβεβαιώνει την παραπάνω εξίσωση. Το διάνυσμα \mathbf{u} αποτελείται από τις συντεταγμένες του κέντρου του υπό εξέταση partition, ενώ ο πίνακας \mathbf{A} , έχει όλα τα στοιχεία του μηδενικά εκτός από αυτά που αντιστοιχούν στην περιοχή στην οποία ανήκει το εκάστοτε κελί.

Ο Gordian χρησιμοποιεί το partitioning για να περιορίσει την ελευθερία κίνησης των κελιών, και όχι για να ελαττώσει το μέγεθος του προβλήματος. Ένας άτυπος περιορισμός για το partitioning είναι πως οι δύο νέες περιοχές, που δημιουργούνται στο εκάστοτε βήμα, πρέπει να έχουν αναλογία $\frac{1}{2}$ στην χειρότερη των περιπτώσεων.

Το final placement είναι, ίσως, το σημαντικότερο βήμα του Gordian. Στις standard cell σχεδιάσεις τα κελιά έχουν το ίδιο ύψος, το μήκος τους, όμως, μπορεί να διαφέρει αισθητά. Ο στόχος είναι να δημιουργήσουμε σειρές στις οποίες θα τοποθετηθούν τα κελιά, οι οποίες θα απέχουν την μικρότερη δυνατή απόσταση μεταξύ τους.

Επιπρόσθετα, το μήκος συμπληρωμένων από κελιά σειρών δεν πρέπει να διαφέρει πάνω από 1-5%, για να έχουμε αισθητή βελτίωση στο τελευταίο στάδιο εφαρμογής του αλγορίθμου.

2.2.2.2 GORDIAN – L

Ο αλγόριθμος Gordian-L ακολουθεί την βασική στρατηγική του Gordian, τροποποιεί μερικώς, όμως, το global placement και τον τρόπο που γίνεται το partitioning. Οι βασικότερες τροποποιήσεις είναι η μετατροπή της αντικειμενικής συνάρτησης σε γραμμική και ο επαναληπτικός διαχωρισμός της εκάστοτε περιοχής σε ίσους χώρους. Η σύγκριση των αποτελεσμάτων της εφαρμογής των δύο αλγορίθμων μας οδηγεί στο συμπέρασμα πως ο Gordian-L παρουσιάζει ελαφρώς καλύτερα αποτελέσματα ως προς την ελαχιστοποίηση της απαιτούμενης περιοχής σε κυκλώματα με μικρό αριθμό πυλών, ενώ ο Gordian παρουσιάζει καλύτερα αποτελέσματα ως προς το συνολικό μήκος καλωδίου σε κυκλώματα με μεγάλο αριθμό πυλών.

2.2.3 FORCED – BASED METHODS

Οι forced-based μέθοδοι είναι κατ' ουσίαν μια υποκατηγορία της χωροθέτησης πυλών με χρήση αναλυτικών μεθόδων. Το όνομά τους προκύπτει από την διαφορετική ερμηνεία του προβλήματος του quadratic placement. Πιο συγκεκριμένα, η ονομασία τους προκύπτει εάν θεωρήσουμε τα στοιχεία του κυκλώματος ως αντικείμενα και τα nets ως ελάσματα που τα συνδέουν. Με βάση αυτή την τροποποίηση το ζητούμενο του προβλήματος, η ελαχιστοποίηση δηλαδή του συνολικού μήκους καλωδίου, μετατρέπεται πλέον στην εύρεση του κατάλληλου τρόπου χωροθέτησης που θέτει το σύστημα που δημιουργείται σε κατάσταση ισορροπίας.

2.2.4 SIMULATED ANNEALING

Η μέθοδος του simulated annealing, παρόλο που χαρακτηρίζεται από την απλότητά της, χρησιμοποιείται σπανίως στην διαδικασία του global placement, όταν δεν έχουν παραχθεί τα επιθυμητά αποτελέσματα, κυρίως λόγω του ότι ο συνολικός χώρος λύσεων είναι υπερβολικά μεγάλος. Παρόλαυτα, χρησιμοποιείται σε ένα ευρέως διαδεδομένο placer, τον Dragon, στο στάδιο του global placement, χωρίς να δίνονται συγκεκριμένες λεπτομέρειες για τον τρόπο εφαρμογής της στην επίσημη περιγραφή του συγκεκριμένου εργαλείου.

2.2.5 CLUSTERING

Μία διαφορετική προσέγγιση πάνω στο πρόβλημα του placement, είναι η ομαδοποίηση των κελιών του κυκλώματος σε συστάδες (clustering). Το clustering παρουσιάζει δύο βασικά πλεονεκτήματα:

- i. Βελτιώνει την ταχύτητα εκτέλεσης λόγω της ελάττωσης του συνόλου αντικειμένων που καλούμαστε να χωροθετήσουμε.
- ii. Βοηθάει στην διαφυγή από τοπικά ελάχιστα που μπορούν να αλλοιώσουν την ποιότητα της τελικής λύσης.

Το βασικότερο μειονέκτημα του clustering είναι ο τρόπος επιλογής των κελιών που θα ομαδοποιηθούν. Οι δύο βασικές προσεγγίσεις είναι είτε μια προεπιλογή που βασίζεται στον αρχικό υπεργράφο είτε η επιλογή βάσει ενός αρχικού placement. Σημαντικότερο παράδειγμα αυτής της κατηγορίας είναι ο αλγόριθμος Timberwolf, ο οποίος χρησιμοποιεί ένα συνδυασμό simulated annealing και clustering.

2.3 FINAL PLACEMENT

Το υποπρόβλημα του final placement περιορίζεται κατ' ουσίαν στην εύρεση του κατάλληλου τρόπου αναπροσαρμογής των θέσεων των κελιών έτσι ώστε να τοποθετηθούν στις βέλτιστες δυνατές θέσεις εξαλείφοντας τυχόν φαινόμενα επικάλυψης. Στα κεφάλαια που ακολουθούν θα παρουσιαστούν οι βασικότερες προσεγγίσεις που άπτονται του προκείμενου προβλήματος.

2.3.1 SIMULATED ANNEALING

Η διαδικασία του simulated annealing εκκινεί με έναν τυχαίο αρχικό διαχωρισμό του κυκλώματος. Ακολούθως, πραγματοποιείται ένας νέος διαχωρισμός των στοιχείων του κυκλώματος ανταλλάσσοντας τις θέσεις ορισμένων κελιών των νέων περιοχών που έχουν δημιουργηθεί. Για κάθε μια από τις αλλαγές αυτές, υπολογίζεται κάποιο κόστος. Εάν το κόστος της κίνησης των κελιών είναι αρνητικό τότε η προκείμενη κίνηση γίνεται δεκτή, εάν είναι ίσο με το μηδέν ή θετικό τότε η κίνηση αυτή γίνεται δεκτή υπό προϋποθέσεις. Αυτή η διαφοροποίηση είναι και ο βασικός λόγος που το simulated annealing μπορεί να υπερβεί τοπικά ελάχιστα στην διαδικασία εύρεσης του καθολικού ελάχιστου.

Στην εικόνα που ακολουθεί παρουσιάζεται μια περιγραφή σε ψευδοκώδικα της διαδικασίας του simulated annealing:

Simulated Annealing Procedure:

```
t := t0;
cur_part := init_part;
cur_score := SCORE( cur_part );
repeat
  repeat
    comp1 := SELECT( part1 );
    comp2 := SELECT( part2 );
    trial_part := EXCHANGE( comp1, comp2, cur_part );
    trial_score := SCORE( trial_part );
    ds := trial_score - cur_score;

    if ( ds < 0 ) then
      cur_score := trial_score;
      cur_part := MOVE( comp1, comp2 );
    else
      r := RANDOM( 0, 1 );
      if( r < e-ds/t ) then
        cur_score := trial_score;
        cur_part := MOVE( comp1, comp2 );
  until( equilibrium )
  t := a*t;    //( 0 < a < 1 )
until( t near 0 )
end_procedure;
```

Η συνάρτηση SELECT χρησιμοποιείται για την επιλογή δύο τυχαίων κελιών, ένα από κάθε partition που σχηματίζεται. Τα κελιά αυτά είναι υποψήφια προς ανταλλαγή θέσης.

Η συνάρτηση EXCHANGE που καλείται στη συνέχεια δημιουργεί δύο δοκιμαστικά partition χωρίς να εκτελεί την ανταλλαγή.

Η συνάρτησης SCORE υπολογίζει το κόστος της κίνησης αυτής, και εάν το κόστος μειώνεται τότε και μόνο τότε πραγματοποιείται η ανταλλαγή κελιών μέσω της κλήσης της MOVE. Το κόστος που υπολογίζεται μπορεί να είναι είτε το cutsize του partitioning, είτε το cutsize σε συνάρτηση με άλλους παράγοντες που έχουμε θέσει προς βελτιστοποίηση.

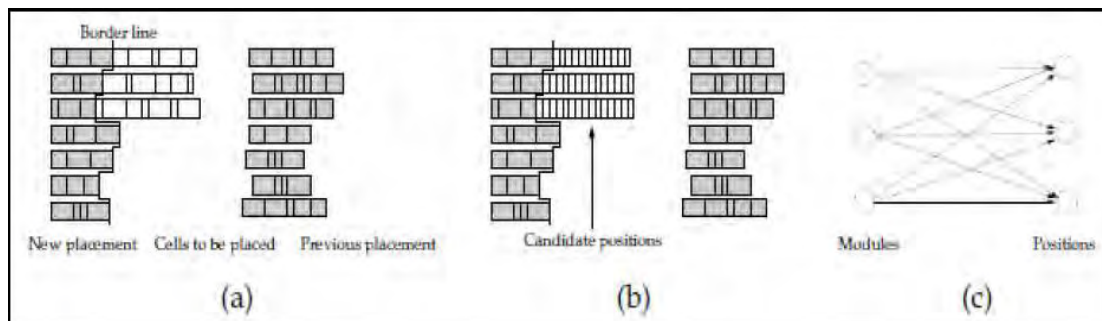
Στην περίπτωση που το τρέχων κόστος είναι μεγαλύτερο από το κόστος πριν από την επιλογή των προς ανταλλαγή κελιών, υπολογίζεται η πιθανότητα αποδοχής του συγκεκριμένου ζεύγους με χρήση της συνάρτησης RANDOM. Εάν η κίνηση γίνει δεκτή καλείται και πάλι η συνάρτηση MOVE για να εκτελέσει την ανταλλαγή. Η ποιότητα της λύσης που παράγεται από την εφαρμογή του simulated annealing εξαρτάται από την επιλογή της μεταβλητής t που είθισται να αναπαριστά την θερμοκρασία και το βήμα μείωσης της προκείμενης θερμοκρασίας (την μεταβλητή a εν προκειμένω). Όσο μεγαλύτερη είναι η αρχική θερμοκρασία και όσο μικρότερος είναι ο ρυθμός ελάττωσής της, τόσο καλύτερα είναι τα αποτελέσματα που προκύπτουν.

Το μειονέκτημα, όμως, είναι πως ο χρόνος που απαιτείται για τον σχηματισμό της κατάλληλης λύσης είναι ανάλογος του συνολικού αριθμού βημάτων μείωσης της θερμοκρασίας.

2.3.2 GREEDY APPROACHES

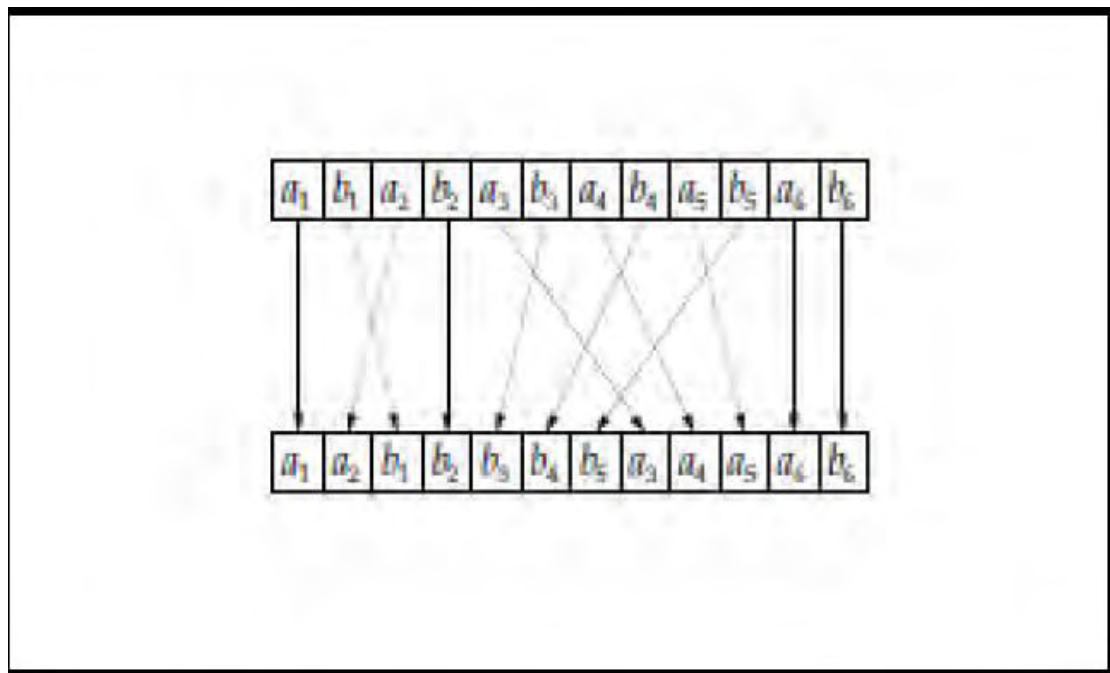
Ένας από τους πιο γνωστούς και ευρέως χρησιμοποιούμενους αλγόριθμους, που ακολουθούν μια άπληστη προσέγγιση ως προς τον τρόπο χωροθέτησης των κελιών είναι ο Domino. Ο αλγόριθμος δέχεται ως είσοδο ένα placement, δεν απαιτεί, όμως, η προκείμενη χωροθέτηση να είναι legalized. Ακολουθώντας, διαχωρίζει τον συνολικό χώρο σε υποπεριοχές, βάσει συγκεκριμένων κριτηρίων, και τις εξετάζει μία προς μία. Κατά τη διάρκεια τροποποίησης της εκάστοτε περιοχής οι υπόλοιπες θεωρούνται ανενεργές. Στην επιλεγμένη περιοχή, ο Domino εκκινεί την εξέταση των κελιών από αριστερά προς τα δεξιά. Κάθε κελί διαχωρίζεται σε μια ομάδα υποκελιών, ίσου μεγέθους, για τα οποία υπολογίζονται νέες «νόμιμες» θέσεις σε κάθε μια από τις προκαθορισμένες σειρές της περιοχής. Βάσει αυτών τον βημάτων διατυπώνεται, σε μαθηματική μορφή, ένα transportation problem. Το κόστος της κάθε μεταφοράς σχετίζεται άμεσα με το netlength του εκάστοτε υποκελιού. Την επίλυση του παραπάνω προβλήματος ακολουθεί η επιλογή της νέας θέσης του κελιού. Η θέση αυτή ταυτίζεται με τη θέση όπου έχουν τοποθετηθεί τα περισσότερα υποκελιά του. Το βασικό πλεονέκτημα του Domino είναι πως λόγω της επαναληπτικής διαδικασίας που ακολουθείται παρουσιάζει την ιδιότητα να αποφεύγει τοπικά ελάχιστα που

μπορούν να εμφανιστούν και να αλλοιώσουν την ποιότητα της τελικής λύσης. Στην εικόνα που ακολουθεί παρουσιάζεται η χρήση του αλγορίθμου για την εύρεση του κατάλληλου placement μιας υποπεριοχής.



Μια, ακόμη, ενδιαφέρουσα προσέγγιση είναι και ο αλγόριθμος Dragon. Ο προκείμενος αλγόριθμος αποδέχεται την global λύση ως ιδανική, οπότε αντί της χρήσης simulated annealing εφαρμόζει μια άπληστη προσέγγιση επίλυσης του προβλήματος χωροθέτησης. Για κάθε ένα από τα κελιά ερευνούνται τα γειτονικά του, έτσι ώστε να γίνει η επιλογή του καταλληλότερου προς ανταλλαγή κελιού. Το σύνολο των γειτονικών κελιών είναι σχετικά μικρό με αποτέλεσμα ο αριθμός των προς ανταλλαγή κελιών να είναι μικρός.

Τέλος, κρίνεται σκόπιμη η αναφορά στον placer Mongrel. Ο συγκεκριμένος placer ασχολείται μόνο με την μετακίνηση των κελιών εντός του εκάστοτε row. Διατηρεί την σχετική σειρά των κελιών και προσπαθεί να βρει τις κατάλληλες υποακολουθίες αυτής που μπορούν να αλλάξουν εκατέρωθεν θέσεις. Η εύρεση του ζεύγους υποακολουθιών έχει αποδειχθεί πως μπορεί να καταστεί δυνατή με χρήση δυναμικού προγραμματισμού εντός πολυωνυμικού χρόνου, αναλόγου με τον συνολικό αριθμό των pins των προς εξέταση κελιών. Στην εικόνα που ακολουθεί παρουσιάζεται η βελτίωση που εμφανίζεται μέσω της εφαρμογής του αλγορίθμου.



2.4 LEGALIZATION

Οι αλγόριθμοι που παρουσιάστηκαν στην παραπάνω ενότητα δεν δίνουν πάντοτε στην έξοδό τους ένα legal placement. Γι' αυτό και στην συνολική διαδικασία χωροθέτησης ενός κυκλώματος εισάγεται και ένα τελικό βήμα «νομιμοποίησης» της σχεδίασης.

Στην περίπτωση του standard-cell placement η διαδικασία που ακολουθείται, στην πλειοψηφία των περιπτώσεων, είναι να γίνεται ένας αρχικός διαχωρισμός των κελιών ως προς τις συντεταγμένες x και y , και ακολούθως να τοποθετούνται στις σειρές που έχουν προκαθοριστεί χρησιμοποιώντας την συντεταγμένη y για την επιλογή της κατάλληλης σειράς και την συντεταγμένη x για την θέση τους εντός της προκείμενης σειράς.

ΚΕΦΑΛΑΙΟ 3

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (ARTIFICIAL INTELLIGENCE - AI)

3.1 ΕΙΣΑΓΩΓΗ

Τεχνητή Νοημοσύνη (Artificial Intelligence AI) είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφυών (νοημόνων) υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζουμε με τη νοημοσύνη στην ανθρώπινη συμπεριφορά (Barr & Feigenbaum). Στόχος της τεχνητής νοημοσύνης είναι να δημιουργηθούν συστήματα που:

- Σκέφτονται όπως οι άνθρωποι.
- Συμπεριφέρονται όπως οι άνθρωποι.
- Σκέφτονται λογικά.
- Αντιδρούν λογικά.

Ουσιαστικά είναι ο τομέας της επιστήμης των υπολογιστών που ασχολείται με τη σχεδίαση και την υλοποίηση προγραμμάτων τα οποία είναι ικανά να μιμηθούν τις ανθρώπινες γνωστικές ικανότητες, εμφανίζοντας έτσι χαρακτηριστικά που αποδίδουμε συνήθως σε ανθρώπινη συμπεριφορά, όπως η επίλυση προβλημάτων, η αντίληψη μέσω της όρασης, η μάθηση, η εξαγωγή συμπερασμάτων και η κατανόηση φυσικής γλώσσας.

3.2 ΠΕΡΙΟΧΕΣ ΤΗΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Η τεχνητή νοημοσύνη θέτει συνεχώς υψηλότερους στόχους και πλέον προσπαθεί να δημιουργήσει συστήματα που εξαρτώνται λιγότερο από τον προγραμματιστή και περισσότερο από την ικανότητα τους να μαθαίνουν πως να συμπεριφέρονται, αλληλεπιδρώντας με το περιβάλλον.

Οι βασικές περιοχές της τεχνητής νοημοσύνης είναι οι ακόλουθες:

- Επίλυση προβλημάτων.
- Απόδειξη θεωρημάτων.
- Επεξεργασία φυσικής γλώσσας.
- Τεχνητή όραση.
- Μηχανική μάθηση (Machine Learning).
- Σχεδιασμός ενεργειών και χρονοπρογραμματισμός.
- Αυτόνομα Robot.
- Έμπειρα συστήματα και συστήματα γνώσης.
- Ευφυείς πράκτορες (agents).
- Ευφυείς υπηρεσίες διαδικτύου και σημασιολογικό διαδίκτυο (semantic web).
- Προσαρμοζόμενα και εξελισσόμενα ευφυή συστήματα.

3.3 ΠΡΟΣΕΓΓΙΣΕΙΣ ΓΙΑ ΤΗΝ ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

Υπάρχουν δύο βασικοί τρόποι προσέγγισης της τεχνητής νοημοσύνης. Η κλασσική ή συμβολική και η υπολογιστική ή μη – συμβολική νοημοσύνη.

Κλασσική ή συμβολική (symbolic AI) : Βασίζεται στην κατανόηση των νοητικών διεργασιών και ασχολείται κυρίως με την προσομοίωση της ανθρώπινης νοημοσύνης προσεγγίζοντάς την με αλγόριθμους και συστήματα που βασίζονται στη γνώση χρησιμοποιώντας ως δομικές μονάδες τα σύμβολα (π.χ. συστήματα κανόνων)

Υπολογιστική νοημοσύνη (computational intelligence) ή Συνδεδετική (connectionist) ή μη – συμβολική : Βασίζεται στη μίμηση της βιολογικής λειτουργίας του εγκεφάλου όπως η διαδικασία της εξέλιξης των ειδών ή η λειτουργία του εγκεφάλου (π.χ. Νευρωνικά δίκτυα, γενετικοί αλγόριθμοι).

3.4 Η ΕΞΕΛΙΞΗ ΤΗΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Αυτή την εποχή βιώνουμε τη μετα – μοντέρνα περίοδο στην οποία η τεχνητή νοημοσύνη καλείται να παίξει ένα σημαντικό ρόλο σε ένα νέο πληροφοριακό περιβάλλον του οποίου κύρια χαρακτηριστικά είναι η εξάπλωση του διαδικτύου και η διείσδυση των υπολογιστικών συστημάτων σε κάθε είδους συσκευές ευρείας και καθημερινής χρήσης (pervasive computing). Αυτή τη στιγμή υπάρχουν συστήματα που μπορούν να βοηθήσουν το χρήστη στο να χρησιμοποιήσει διάφορα προγράμματα (assistants), να αναζητήσει πληροφορία στο διαδίκτυο, να στείλει mail, να τηρήσει τα ραντεβού του και να συγκρίνει τιμές διάφορων προϊόντων βρίσκοντας την καλύτερη λύση. Υπάρχουν συστήματα αναγνώρισης φωνής που σου δίνουν τη δυνατότητα να κλείσεις αεροπορικές θέσεις τηλεφωνικά, επιλέγοντας τις βέλτιστες πτήσεις ή ακόμη να σε πληροφορήσουν για τη διάρκεια και τις τιμές αυτών. Έμπειρα συστήματα πραγματικού χρόνου επεξεργάζονται τα δεδομένα που μεταδίδονται από τα διαστημόπλοια. Ρομποτικά συστήματα που οδηγούν αυτοκίνητα σε αυτοκινητόδρομο χρησιμοποιώντας video κάμερες και sonar ή ακόμη και να ελέγχουν και να ρυθμίζουν την κυκλοφορία των αυτοκινήτων. Η διεξαγωγή ιατρικών εξετάσεων και διαγνώσεων είναι ακόμη ένα βασικό χαρακτηριστικό αυτών των συστημάτων.

Γνωστές και μεγάλες εταιρίες τεχνολογίας έχουν αναπτύξει αυτόνομα ρομπότ. Χαρακτηριστικά παραδείγματα το σκυλάκι AIBO με δυνατότητα αναγνώρισης ομιλίας, έκφρασης συναισθημάτων με λόγο ή κινήσεις και το new generation ρομπότ, QRIO, το οποίο μπορεί να χορέψει και να επικοινωνήσει αναγνωρίζοντας 10.000 λέξεις, μεταξύ αυτών και τα ελληνικά.

ΚΕΦΑΛΑΙΟ 4

ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ (MACHINE LEARNING)

4.1 ΕΙΣΑΓΩΓΗ

Με τον όρο της μηχανικής μάθησης (machine learning) αναφερόμαστε στη μάθηση σε ένα γνωστικό σύστημα, όπως μπορεί αυτή να γίνει αντιληπτή στην καθημερινή ζωή. Μπορεί να συνδεθεί με δύο βασικές ιδιότητες. Την ικανότητα στην πρόσκτηση γνώσης κατά την αλληλεπίδραση του με το περιβάλλον, και στην ικανότητα να βελτιώνει με την επανάληψη τον τρόπο εκτέλεσης μίας ενέργειας. Ως μηχανική μάθηση, λοιπόν, προσδιορίζεται η πρόσκτηση επιπλέον γνώσης, που επιφέρει μεταβολές στην υπάρχουσα γνώση.

Όλοι μας προσπαθούμε να κατανοήσουμε το περιβάλλον γύρω μας παρατηρώντας το και δημιουργώντας μια απλοποιημένη (αφαιρετική) εκδοχή του που ονομάζεται μοντέλο (model). Η δημιουργία ενός τέτοιου μοντέλου, ονομάζεται επαγωγική μάθηση (inductive learning) ενώ η διαδικασία γενικότερα ονομάζεται επαγωγή (induction). Εμείς έχουμε τη δυνατότητα να οργανώνουμε και να συσχετίζουμε τις εμπειρίες και τις παραστάσεις του δημιουργώντας νέες δομές που ονομάζονται πρότυπα (patterns).

Η δημιουργία μοντέλων ή προτύπων από ένα σύνολο δεδομένων, από ένα υπολογιστικό σύστημα είναι αυτό που ονομάζουμε μηχανική μάθηση machine learning).

4.2 ΕΙΔΗ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Έχουν αναπτυχθεί πολλές τεχνικές μηχανικής μάθησης που χρησιμοποιούνται ανάλογα με τη φύση του προβλήματος. Αυτές οι τεχνικές εμπίπτουν κυρίως σε ένα από τα παρακάτω είδη:

- ο *Μάθηση με επίβλεψη (supervised learning)* : Το σύστημα καλείται να “μάθει” μια έννοια ή συνάρτηση από ένα σύνολο δεδομένων, η οποία αποτελεί περιγραφή ενός μοντέλου.

- ο *Μάθηση χωρίς επίβλεψη (unsupervised learning)* : Το σύστημα πρέπει μόνο του να ανακαλύψει συσχετίσεις ή ομάδες σε ένα σύνολο δεδομένων, δημιουργώντας πρότυπα, χωρίς να είναι γνωστό αν υπάρχουν, πόσα και ποια είναι.

4.3 ΜΑΘΗΣΗ ΜΕ ΕΠΙΒΛΕΨΗ

Στη μάθηση με επίβλεψη το σύστημα πρέπει να “μάθει” επαγωγικά μια συνάρτηση που ονομάζεται συνάρτηση στόχος (target function) και αποτελεί έκφραση του μοντέλου που περιγράφει τα δεδομένα. Η συνάρτηση στόχος χρησιμοποιείται για την πρόβλεψη της τιμής μιας μεταβλητής, που ονομάζεται εξαρτημένη μεταβλητή ή μεταβλητή εξόδου, βάσει των τιμών ενός συνόλου μεταβλητών, που ονομάζονται ανεξάρτητες μεταβλητές ή μεταβλητές εισόδου ή χαρακτηριστικά.

Η επαγωγική μάθηση στηρίζεται στην “υπόθεση επαγωγικής μάθησης” (inductive learning hypothesis), σύμφωνα με την οποία κάθε υπόθεση h που προσεγγίζει καλά τη συνάρτηση στόχο για ένα αρκετά μεγάλο σύνολο παραδειγμάτων, θα προσεγγίζει το ίδιο καλά τη συνάρτηση στόχο και για περιπτώσεις που δεν έχει εξετάσει.

Στην μάθηση με επίβλεψη διακρίνονται δυο είδη προβλημάτων (learning tasks), τα προβλήματα ταξινόμησης και τα προβλήματα παρεμβολής.

Η *ταξινόμηση - κατηγοριοποίηση (classification)* αφορά στη δημιουργία μοντέλων πρόβλεψης διακριτών τάξεων (κλάσεων/κατηγοριών) (π.χ. ομάδα αίματος).

Η *παρεμβολή (regression)* αφορά στη δημιουργία μοντέλων πρόβλεψης αριθμητικών τιμών (π.χ. πρόβλεψη ισοτιμίας νομισμάτων ή τιμής μετοχής).

4.3.1 ΚΑΝΟΝΕΣ ΤΑΞΙΝΟΜΗΣΗΣ

Οι κανόνες if-then είναι από τις πιο εκφραστικές και κατανοητές για τον άνθρωπο αναπαραστάσεις. Ως κυριότερες κατηγορίες κανόνων μπορούμε να θεωρήσουμε τους προτασιακούς (propositional rules) και τους κατηγορηματικούς κανόνες πρώτης τάξης (first order predicate rules).

Οι *προτασιακοί κανόνες* μπορεί να προκύψουν από άλλες μορφές αναπαράστασης (π.χ. δένδρα, γενετικούς αλγόριθμους) αλλά και από απ' ευθείας μάθηση με αλγόριθμους σειριακής κάλυψης. Δεν περιλαμβάνουν μεταβλητές και έτσι δεν μπορεί να αναπαραστήσουν γενικές σχέσεις ανάμεσα στις τιμές των χαρακτηριστικών.

Οι *κατηγορηματικοί κανόνες* πρώτης τάξης περιέχουν μεταβλητές (μεγάλη εκφραστική ικανότητα). Προκύπτουν με απ' ευθείας μάθηση μέσω αλγορίθμων μάθησης κανόνων 1ης τάξης.

4.3.2 ΠΑΡΕΜΒΟΛΗ – ΠΑΛΙΝΔΡΟΜΗΣΗ (REGRESSION)

Πρόκειται για μια διαδικασία προσδιορισμού της σχέσης μιας μεταβλητής y (εξαρτημένη μεταβλητή ή έξοδος) με μια ή περισσότερες άλλες μεταβλητές x_1, x_2, \dots, x_n (ανεξάρτητες μεταβλητές ή εισόδοι). Σκοπός της είναι η πρόβλεψη της τιμής της εξόδου όταν είναι γνωστές οι εισόδοι. Το πιο διαδεδομένο μοντέλο είναι το γραμμικό (linear) όπου η αναμενόμενη τιμή της εξόδου μοντελοποιείται με μία γραμμική συνάρτηση ή σταθμισμένο άθροισμα (weighted sum) των παραμέτρων εισόδου.

4.4 ΜΑΘΗΣΗ ΧΩΡΙΣ ΕΠΙΒΛΕΨΗ

Στη μάθηση χωρίς επίβλεψη το σύστημα έχει στόχο να ανακαλύψει συσχετίσεις και ομάδες από τα δεδομένα, βασιζόμενο μόνο στις ιδιότητές τους. Σαν αποτέλεσμα προκύπτουν πρότυπα (περιγραφές), κάθε ένα από τα οποία περιγράφει ένα μέρος από τα δεδομένα. Βασικά παραδείγματα προτύπων πληροφόρησης είναι οι κανόνες συσχέτισης (*association rules*) και οι ομάδες (*clusters*), οι οποίες προκύπτουν από τη διαδικασία της ομαδοποίησης (*clustering*).

4.4.1 ΚΑΝΟΝΕΣ ΣΥΣΧΕΤΙΣΗΣ

Η ανακάλυψη ή εξόρυξη κανόνων συσχέτισης (*association rule mining*) εμφανίστηκε αρκετά αργότερα από τη μηχανική μάθηση και έχει περισσότερες επιρροές από την ερευνητική περιοχή των βάσεων δεδομένων. Προτάθηκε στις αρχές της δεκαετίας του '90 από τον Rakesh Agrawal ως τεχνική ανάλυσης καλαθιού αγορών (*market basket analysis*) όπου το ζητούμενο είναι η ανακάλυψη συσχετίσεων ανάμεσα στα αντικείμενα μιας βάσης δεδομένων. Στο συγκεκριμένο πρόβλημα υπάρχει ένας μεγάλος αριθμός αντικειμένων (*items*), για παράδειγμα ψωμί, γάλα, κτλ. Οι πελάτες γεμίζουν τα καλάθια τους με κάποιο υποσύνολο αυτών των αντικειμένων και το ζητούμενο είναι να βρεθεί ποια από αυτά τα αντικείμενα αγοράζονται μαζί, χωρίς να ενδιαφέρει ποιος είναι ο αγοραστής.

Οι κανόνες συσχέτισης είναι προτάσεις της μορφής $\{X_1, \dots, X_n\} \rightarrow Y$, που σημαίνει ότι αν βρεθούν όλα τα X_1, \dots, X_n στο καλάθι (στην ανάλυση καλαθιού αγορών) τότε είναι πιθανό να βρεθεί και το Y .

Απλή αναφορά ενός τέτοιου κανόνα δεν έχει μεγάλη αξία αν δε συνοδεύεται από κάποια ποσοτικά μεγέθη που μετρούν την ποιότητα των ευρεθέντων κανόνων συσχέτισης. Τέτοια μεγέθη είναι :

- ο Υποστήριξη (*support*) ή κάλυψη (*coverage*) που εκφράζει την πιθανότητα να βρεθεί το καλάθι $\{X_1, \dots, X_n, Y\}$ στη βάση δεδομένων και ισούται με το λόγο των εγγραφών που περιλαμβάνουν το $\{X_1, \dots, X_n, Y\}$ προς το σύνολο των εγγραφών.

ο *Εμπιστοσύνη (confidence) ή ακρίβεια (accuracy)*: εκφράζει την πιθανότητα να βρεθεί το Y σε ένα καλάθι που περιέχει τα $\{X_1, \dots, X_n\}$ και ισούται με το λόγο των εγγραφών που περιλαμβάνουν το $\{X_1, \dots, X_n, Y\}$ προς το σύνολο των εγγραφών που περιλαμβάνουν τα X_i .

Για την ανακάλυψη κανόνων συσχέτισης χρησιμοποιείται η ιδιότητα της μονοτονίας (monotonicity property) ή αλλιώς ιδιότητα a priori σύμφωνα με την οποία αν ένα σύνολο αντικειμένων S είναι συχνό, τότε όλα τα υποσύνολα του S είναι επίσης συχνά.

4.4.2 ΟΜΑΔΕΣ (CLUSTERS)

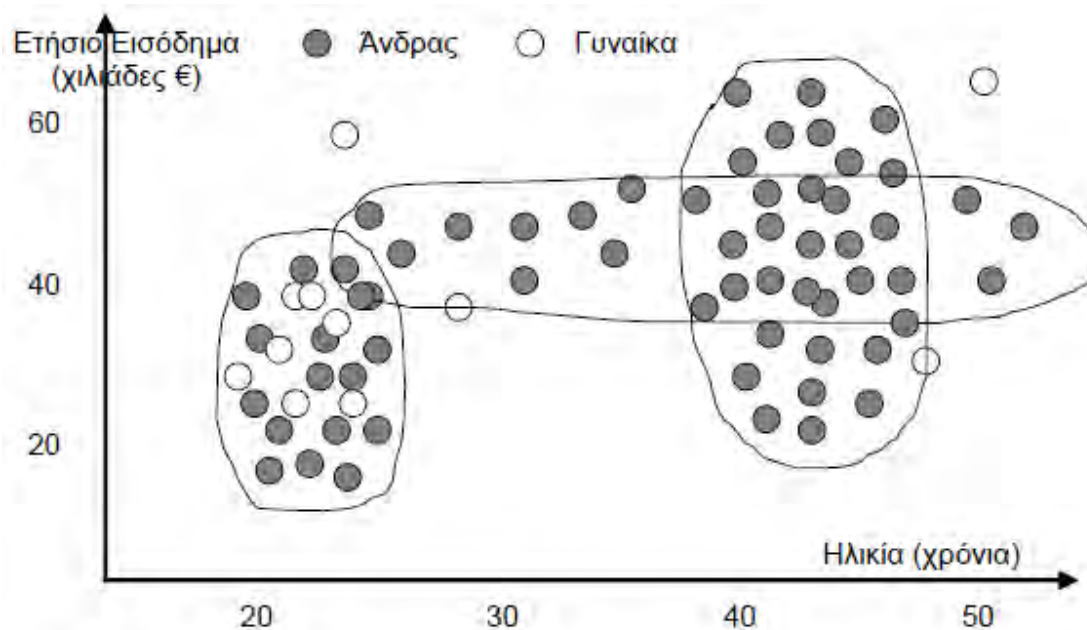
Είναι πρότυπα πληροφόρησης που προκύπτουν με ομαδοποίηση (clustering) δηλαδή διαχωρισμό ενός συνόλου (συνήθως πολυδιάστατων) δεδομένων σε ομάδες, έτσι ώστε τα σημεία που ανήκουν στην ίδια ομάδα να μοιάζουν όσο το δυνατόν περισσότερο και τα σημεία που ανήκουν σε διαφορετικές ομάδες να διαφέρουν όσο το δυνατόν περισσότερο.

Στο Σχήμα που ακολουθεί απεικονίζεται γραφικά μία υποθετική ομαδοποίηση σε δεδομένα αγοραστών σπορ αυτοκινήτων, με βάση την ηλικία (άξονας x), το ετήσιο εισόδημα (άξονας y) και το φύλο. Διακρίνονται τρεις ομάδες:

- αγοραστές νεαρής ηλικίας ανεξαρτήτως φύλλου,
- άνδρες αγοραστές με υψηλό εισόδημα, όλων των ηλικιών μέχρι τα 53 χρόνια

και

- άνδρες αγοραστές ηλικίας περίπου 44 ανεξαρτήτως εισοδήματος.



Υπάρχουν τρεις γενικές κατηγορίες αλγορίθμων ομαδοποίησης:

- Οι αλγόριθμοι βασισμένοι σε διαχωρισμούς (*partition based*), που προσπαθούν να βρουν τον καλύτερο διαχωρισμό ενός συνόλου δεδομένων σε ένα συγκεκριμένο αριθμό ομάδων.
- Οι ιεραρχικοί (*hierarchical*) αλγόριθμοι, που προσπαθούν με ιεραρχικό τρόπο να ανακαλύψουν τον αριθμό και τη δομή των ομάδων.
- Οι πιθανοκρατικοί (*probabilistic*) αλγόριθμοι, που βασίζονται σε μοντέλα πιθανοτήτων.

4.5 ΑΛΛΑ ΕΙΔΗ ΜΑΘΗΣΗΣ

Εκτός από τις μεθόδους που παρουσιάστηκαν, υπάρχουν και άλλες προσεγγίσεις στο πρόβλημα της μηχανικής μάθησης. Δύο από αυτές, είναι οι *γενετικοί αλγόριθμοι* και η *ενισχυτική μάθηση*.

Γενετικοί Αλγόριθμοι (Genetic Algorithms) : Μέθοδος μάθησης που βασίζεται στην προσομοίωση του φυσικού φαινομένου της εξέλιξης (*evolution*). Οι υποθέσεις συνήθως αναπαρίστανται από ακολουθίες bit (*bit-strings*). Η αναζήτηση της κατάλληλης υπόθεσης ξεκινάει τυχαία με έναν πληθυσμό (μια συλλογή) αρχικών

υποθέσεων, τα μέλη του οποίου παράγουν τη νέα "γενιά" μέσω διαδικασιών αναπαραγωγής αντίστοιχων των βιολογικών, όπως διασταύρωση (crossover) ή τυχαία μετάλλαξη (random mutation). Σε κάθε βήμα, οι υποθέσεις του τρέχοντος πληθυσμού αξιολογούνται βάσει μιας προκαθορισμένης συνάρτησης καταλληλότητας (fitness function). Βάσει αυτής επιλέγονται για το αν θα υφίστανται ή όχι στην επόμενη γενιά. Δηλαδή, η μάθηση αντιμετωπίζεται σαν μία ειδική περίπτωση βελτιστοποίησης.

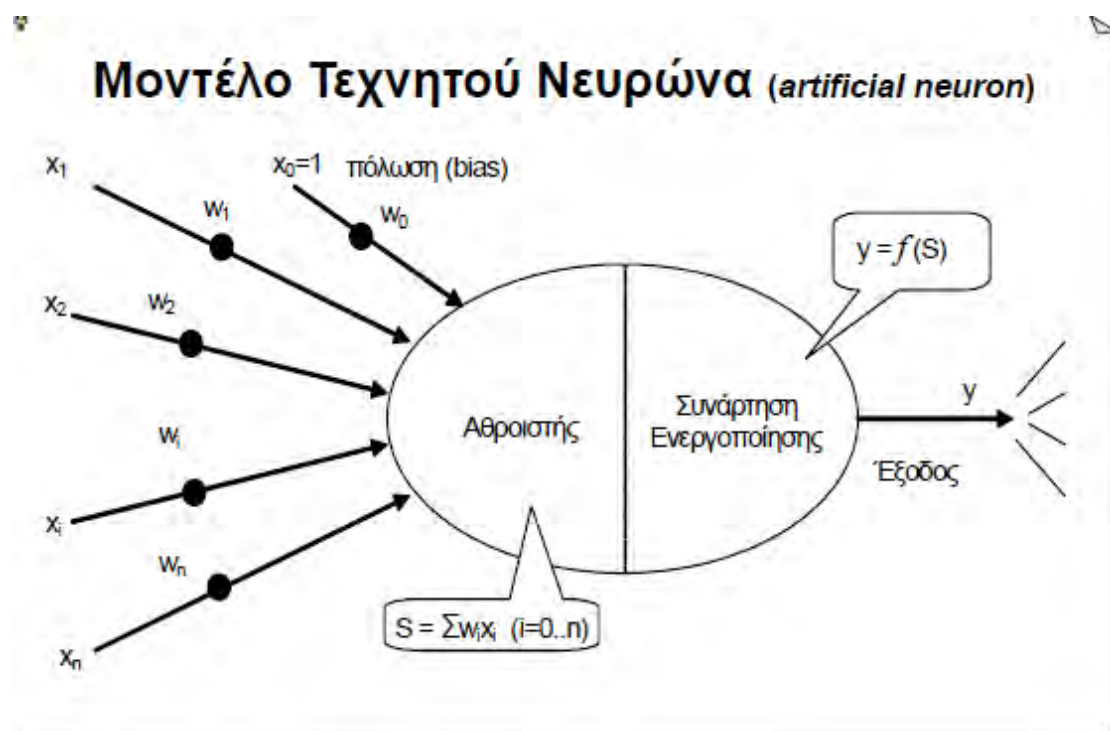
Ενισχυτική Μάθηση (Reinforcement Learning) : Γενική περιγραφή οικογένειας τεχνικών στις οποίες το σύστημα μάθησης προσπαθεί να μάθει μέσω άμεσης αλληλεπίδρασης με το περιβάλλον. Ουσιαστικές εφαρμογές αυτών είναι ο έλεγχος κίνησης ρομπότ, η βελτιστοποίηση εργασιών σε εργοστάσια, η μάθηση επιτραπέζιων παιχνιδιών, κτλ. Είναι εμπνευσμένη από τα αντίστοιχα ανάλογα της μάθησης με επιβράβευση και τιμωρία που συναντώνται στα έμβια όντα. Σκοπός του συστήματος μάθησης είναι να μεγιστοποιήσει μια συνάρτηση του αριθμητικού σήματος ενίσχυσης (ανταμοιβή), για παράδειγμα την αναμενόμενη τιμή του σήματος ενίσχυσης στο επόμενο βήμα. Το σύστημα δεν καθοδηγείται από κάποιον εξωτερικό επιβλέποντα για το ποια ενέργεια θα πρέπει να ακολουθήσει αλλά πρέπει να ανακαλύψει μόνο του ποιες ενέργειες είναι αυτές που θα του αποφέρουν το μεγαλύτερο κέρδος.

ΚΕΦΑΛΑΙΟ 5

ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ (NEURAL NETS)

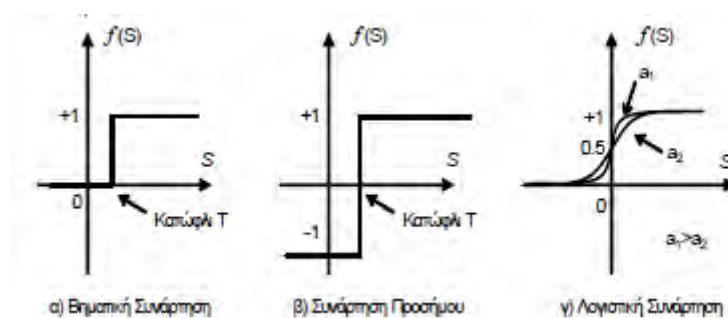
5.1 ΕΙΣΑΓΩΓΗ

Τα Νευρωνικά δίκτυα (που ανήκουν στην μη συμβολική τεχνητή νοημοσύνη) είναι μια ιδιαίτερη προσέγγιση στη δημιουργία συστημάτων με νοημοσύνη. Δεν αναπαριστούν ρητά τη γνώση και δεν υιοθετούν ειδικά σχεσιασμένους αλγόριθμους αναζήτησης. Βασίζονται κυρίως σε βιολογικά πρότυπα, όπως ο ανθρώπινος εγκέφαλος. Έχουν δυνατότητα μάθησης μετασχηματίζοντας την εσωτερική τους δομή, παρά καταχωρώντας κατάλληλα αναπαριστάμενη γνώση. Στον ανθρώπινο εγκέφαλο υπάρχουν ~100 δισεκατομμύρια νευρώνες. Κάθε νευρώνας συνδέεται κατά μέσο όρο με 1000 άλλους νευρώνες, που σημαίνει ότι έχουμε ~100 τρισεκατομμύρια συνάψεις. Ο χρόνος απόκρισης των βιολογικών νευρώνων είναι της τάξης των msec και μέσα σε αυτό το χρονικό διάστημα λαμβάνει περίπλοκες αποφάσεις εκπληκτικά γρήγορα. Η υπολογιστική ικανότητα του εγκεφάλου και η πληροφορία που περιέχει είναι διαμοιρασμένα σε όλο του τον όγκο. Πρόκειται δηλαδή για ένα παράλληλο και κατανεμημένο υπολογιστικό σύστημα.



5.2 ΣΥΝΑΡΤΗΣΕΙΣ ΕΝΕΡΓΟΠΟΙΗΣΗΣ

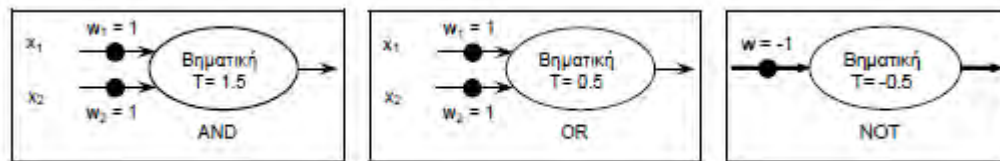
Βασική απαίτηση της συνάρτησης ενεργοποίησης είναι να μην είναι γραμμική ώστε να μπορεί να μοντελοποιεί μη γραμμικά φαινόμενα.



Η λογιστική (logistic) συνάρτηση – μέλος οικογένειας σιγμοειδών συναρτήσεων.

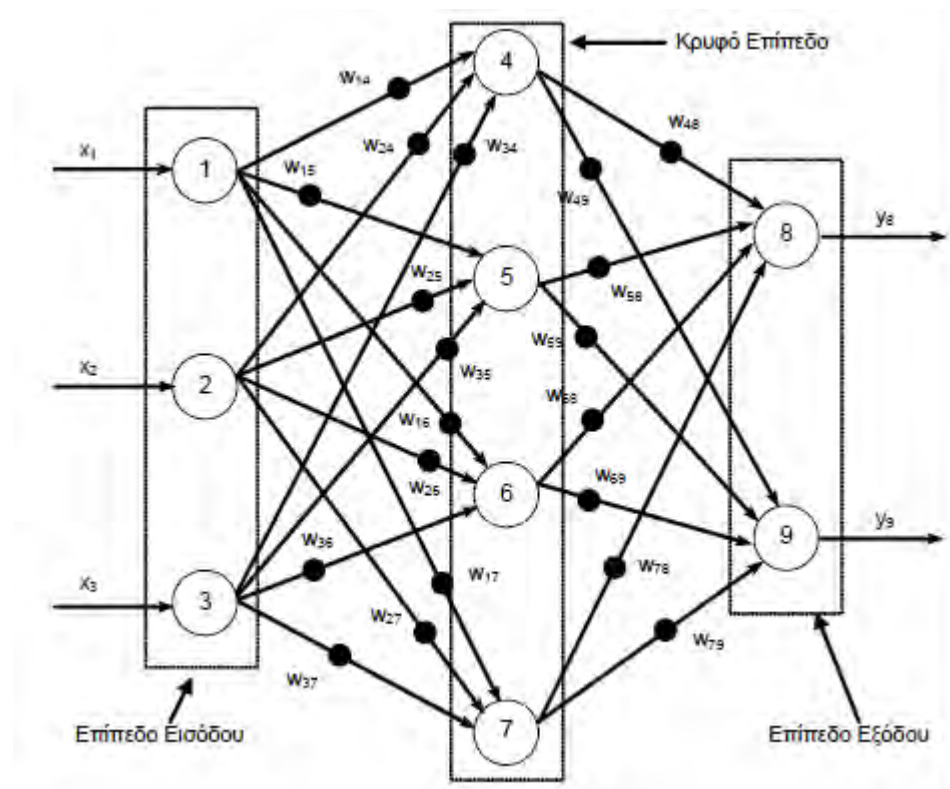
$$\Phi(S) = \frac{1}{1 + e^{-a \cdot S}}$$

Ακολουθεί σχηματικό παράδειγμα υλοποίησης λογικών συναρτήσεων με τεχνητό νευρώνα.



5.3 ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Πρόκειται για συστήματα επεξεργασίας δεδομένων που αποτελούνται από ένα πλήθος τεχνητών νευρώνων οργανωμένων σε δομές παρόμοιες με αυτές του ανθρώπινου εγκεφάλου.



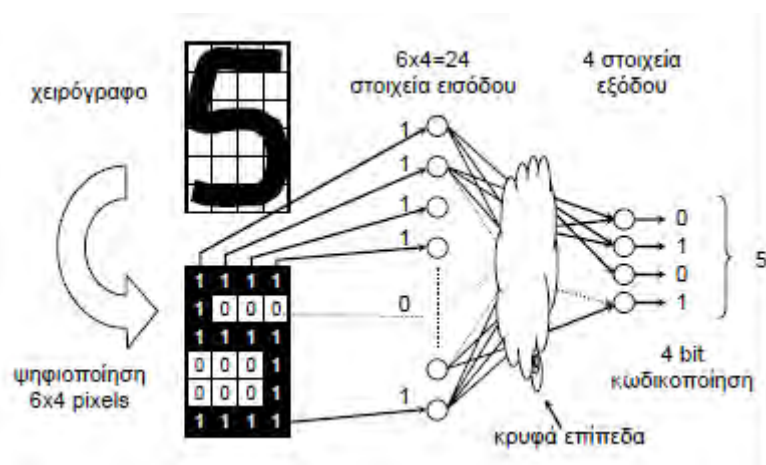
Το παραπάνω νευρωνικό δίκτυο είναι πλήρως συνδεδεμένο με απλή ανατροφοδότηση. Οι νευρώνες των διάφορων στρωμάτων μπορεί να είναι είτε πλήρως συνδεδεμένοι (*fully connected*) είτε μερικώς συνδεδεμένοι (*partially connected*). Τώρα, τα τεχνητά νευρωνικά δίκτυα μπορούν να χαρακτηριστούν είτε ως δίκτυα με πρόσθια ανατροφοδότηση (*feed forward*) είτε δίκτυα με ανατροφοδότηση (*feedback* ή *recurrent*).

Τα τεχνητά νευρωνικά δίκτυα έχουν την ικανότητα να μαθαίνουν μέσω παραδειγμάτων (*learn by example*). Το γεγονός ότι μπορούμε να τα θεωρήσουμε ως κατανεμημένη μνήμη (*distributed memory*) και ως μνήμη συσχέτισης (*associative memory*) αποτελεί τεράστιο πλεονέκτημά τους. Έχουν μεγάλη ανοχή στα σφάλματα (*fault - tolerant*) και εξαιρετική ικανότητα στην αναγνώριση προτύπων (*pattern recognition*).

5.3.1 ΠΡΟΣΘΙΑ ΤΡΟΦΟΔΟΤΗΣΗ

Για την τοπολογία του δικτύου δεν υπάρχει κανόνας σχετικά με τον προσδιορισμό των κρυφών επιπέδων, νευρώνων ανά επίπεδο και συνδεσμολογίας. Τα δεδομένα εισόδου – εξόδου βοηθούν στην εκτίμηση του αριθμού νευρώνων στα επίπεδα εισόδου και εξόδου.

Το είδος της μάθησης που χρησιμοποιείται είναι μηχανική μάθηση με επίβλεψη.



5.4 ΑΛΓΟΡΙΘΜΟΙ ΜΑΘΗΣΗΣ ΥΠΟ ΕΠΙΒΛΕΨΗ

Το υπολογιστικό πρόγραμμα δέχεται τις παραδειγματικές εισόδους καθώς και τα επιθυμητά αποτελέσματα από έναν «δάσκαλο», και ο στόχος είναι να μάθει έναν γενικό κανόνα προκειμένου να αντιστοιχίσει τις εισόδους με τα αποτελέσματα.

Βασικοί αλγόριθμοι μάθησης (supervised learning) :

- ο Κανόνας Δέλτα (Delta rule learning)
- ο Αλγόριθμος ανάστροφης μετάδοσης λάθους (back propagation)
- ο Ανταγωνιστική μάθηση (competitive learning)
- ο Τυχαία μάθηση (random learning)

5.4.1 ΚΑΝΟΝΑΣ ΔΕΛΤΑ

Αποτελεί μια γενίκευση του αλγορίθμου μάθησης του perceptron. Βοηθά στην ελαχιστοποίηση του μέσου τετραγώνου σφάλματος των διανυσμάτων εκπαίδευσης και δεν μπορεί να εφαρμοστεί σε δίκτυα με κρυφά επίπεδα.

Μέσο τετραγωνικό σφάλμα E στο στοιχειώδες perceptron για p διανύσματα εκπαίδευσης :

$$E = \frac{1}{p} \sum_{k=1}^p (t_k - input_k)^2$$

Ως $input_k$ ορίζεται το σήμα εισόδου του νευρώνα χωρίς τη συνάρτηση ενεργοποίησης. Το συνολικό σήμα εισόδου για κάποιο διάστημα p είναι :

$$input = \sum_{i=1}^n w_i x_i$$

Όπου t_k η επιθυμητή έξοδος

Ο κανόνας Δέλτα ακολουθεί την αρνητική κλίση της επιφάνειας σφάλματος, με κατεύθυνση προς το ελάχιστό της :

$$\Delta w_i \propto - \frac{\partial E}{\partial w_i}$$

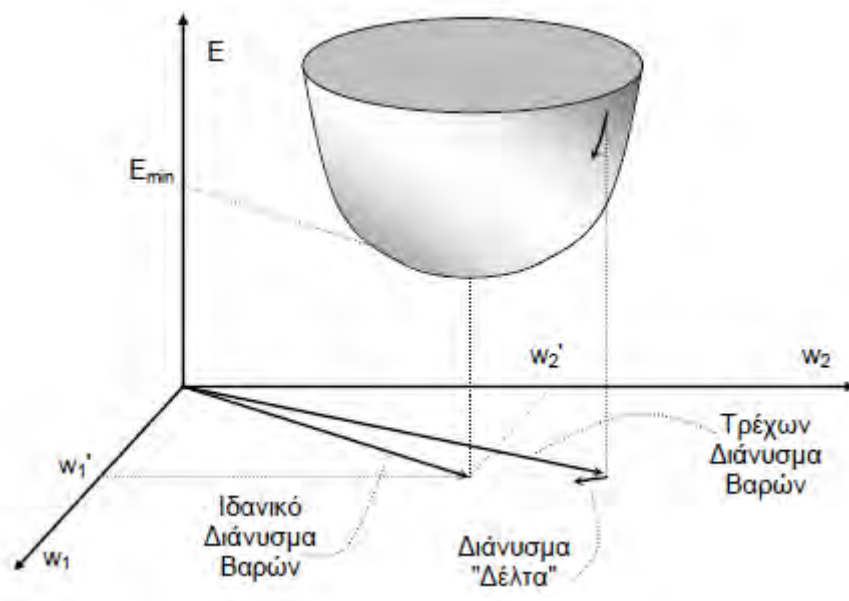
Η παράγωγος του E ως προς το w είναι :

$$\nabla E = \left(\frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right)$$

Η μεταβολή στην τιμή του βάρους w_i εξαιτίας της εκπαίδευσης με ένα μόνο από τα διανύσματα εκπαίδευσης, δίνεται από τη σχέση :

$$\Delta w_i = w_{i(new)} - w_{i(old)} = d(t - input)x_i$$

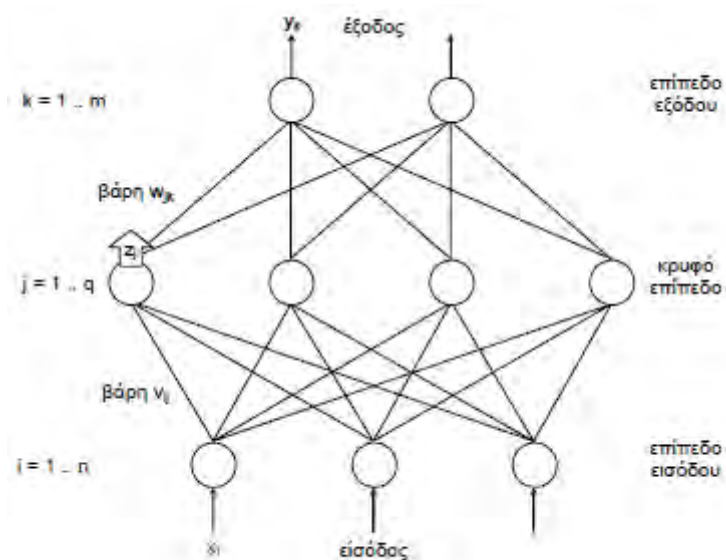
Ακολουθεί γεωμετρική αναπαράσταση κανόνα Δέλτα για στοιχειώδες perceptron με δύο εισόδους. Εφαρμόζεται ο κανόνας της επικλινούς καθόδου (gradient descent rule). Διαπιστώθηκε όμως ότι δεν μπορεί να εφαρμοστεί σε νευρωνικά δίκτυα με κρυφά επίπεδα, γεγονός που περιόρισε την έρευνα των τεχνητών νευρωνικών δικτύων για πολλά χρόνια.



Η λύση στο πρόβλημα δόθηκε με τη μέθοδο της ανάστροφης μετάδοσης του λάθους.

5.4.2 ΑΝΑΣΤΡΟΦΗ ΜΕΤΑΔΟΣΗ ΛΑΘΟΥΣ

Βασίζεται στον γενικευμένο αλγόριθμο Δέλτα (generalized Delta rule). Βασική ιδέα είναι να καθοριστεί το ποσοστό του συνολικού σφάλματος που αντιστοιχεί σε κάθε νευρώνα, ακόμη και αυτών που ανήκουν σε κρυφά επίπεδα.



$$input_j = \sum_{i=1}^n v_{ji} x_i \quad z_j = f(input_j) = f\left(\sum_{i=1}^n v_{ji} x_i\right)$$

$$input_k = \sum_{j=1}^q w_{jk} z_j \quad y_k = f(input_k) = f\left(\sum_{j=1}^q w_{jk} z_j\right)$$

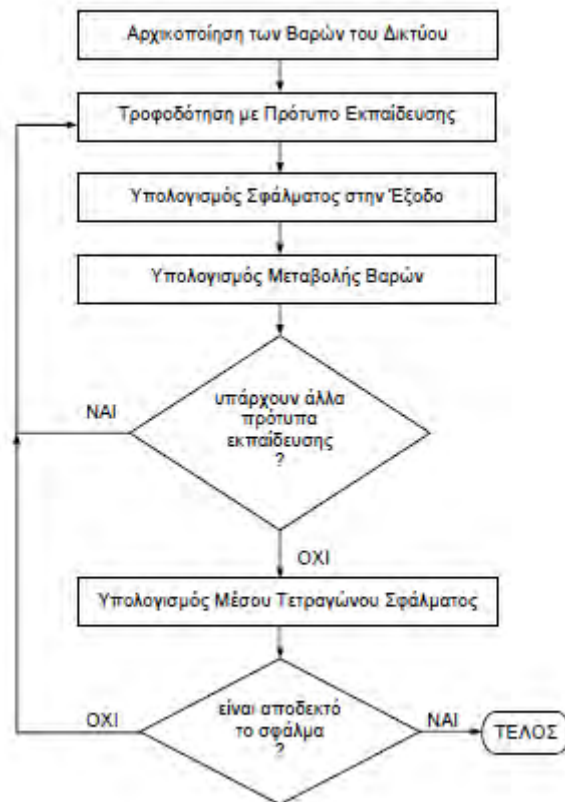
Αποδεικνύεται ότι :

Για επίπεδο εξόδου $\Delta w_{jk} = d \cdot \delta_k \cdot z_j$ με $\delta_k = (t_k - y_k) f'(input_k)$

Για κρυφό επίπεδο $\delta_j = f'(input_j) \sum_{k=1}^m \delta_k w_{jk}$ με $\Delta w_{ji} = d \cdot \delta_j \cdot x_i$

Η αναπαράσταση των βαρών γίνεται από το επίπεδο εξόδου προς το εισόδου, ανάστροφο πέρασμα (backward pass) ή ανάστροφη μετάδοση (back propagation). Χρησιμοποιείται επίσης διαδικασία βελτιστοποίησης επικλινούς καθόδου (gradient descent optimization procedure) που ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα Ε μεταξύ της εξόδου του δικτύου και της επιθυμητής εξόδου, για τα p διανύσματα εκπαίδευσης. Τα διανύσματα εκπαίδευσης δεν πρέπει να μεταβάλλονται διότι δεν επέρχεται σύγκλιση.

Τα βασικότερα προβλήματα είναι τα τοπικά ελάχιστα και η network paralysis, το γεγονός δηλαδή ότι ένα ή περισσότερα βάρη έχουν σταθερά υψηλές απόλυτες τιμές και δεν τροποποιούνται σημαντικά σε κάθε διόρθωση.



ΚΕΦΑΛΑΙΟ 6

ΒΙΒΛΙΟΓΡΑΦΙΑ

- i. <http://ai.stanford.edu/~ang/papers/icml11-MultimodalDeepLearning.pdf>
- ii. Automatic placement : A review of current techniques *Bryan T. Preas, Patrick G. Karger*
- iii. Artificial neural Networks : A tutorial A.K. Jain, Jianchang Mao, K.M. Mohiuddin
- iv. Partitioning Very Large Circuits Using Analytical Placement Techniques B.M. Riess, K. Doll, F.M. Johannes
- v. A universal modular ACTOR formalism for artificial intelligence Carl Hewitt, Peter Bishop, Richard Steiger
- vi. Τεχνητή Νοημοσύνη Β' Έκδοση Βλαχάβας, Κεφαλάς, Βασιλειάδης, Κόκκορας, Σακελλαρίου
- vii. Multilevel generalized force-directed method for circuit placement Tony Chan, Jason Cong, Kenton Sze
- viii. Large-scale circuit placement Jason Cong, Xin Yuan, Tim Kong, Min Xie, Joseph R. Shinnerl